

CENTRO UNIVERSITÁRIO DA FEI  
HORST ERICH ROSA ERDMANN

**DETECÇÃO E CLASSIFICAÇÃO DE IMAGENS SINTÉTICAS UTILIZANDO  
ENTROPIA NÃO-EXTENSIVA**

**São Bernardo do Campo, SP**

**2009**

# **Livros Grátis**

<http://www.livrosgratis.com.br>

Milhares de livros grátis para download.

HORST ERICH ROSA ERDMANN

**DETECÇÃO E CLASSIFICAÇÃO DE IMAGENS SINTÉTICAS UTILIZANDO  
ENTROPIA NÃO-EXTENSIVA**

Dissertação de Mestrado apresentada ao Centro Universitário da FEI para obtenção do título de Mestre em Engenharia Elétrica, orientada pelo Prof. Dr. Paulo Sérgio Rodrigues.

São Bernardo do Campo, SP  
**2009**

Erdmann, Horst Erich Rosa

Detecção e classificação de imagens sintéticas utilizando entropia não-extensiva / Horst Erich Rosa Erdmann. São Bernardo do Campo, 2009.

120 f. : il.

Dissertação de Mestrado - Centro Universitário da FEI.

Orientador: Prof. Dr. Paulo Sérgio Rodrigues

1. Objetos sintéticos. 2. Automação de Tarefas. 3. Entropia. I. Rodrigues, Paulo Sérgio, orient. III. Título.

CDU 681.32



Centro Universitário da FEI

# APRESENTAÇÃO DE DISSERTAÇÃO ATA DA BANCA JULGADORA

PGE- 10

## Programa de Mestrado de Engenharia Elétrica

Aluno: Horst Erich Rosa Erdmann

Matrícula: 1071067

Título do Trabalho: Detecção e Classificação de Imagens Sintéticas utilizando Entropia Não-extensiva.

Área de Concentração: Inteligência Artificial Aplicada à Automação

Orientador: Prof. Dr. Paulo Sérgio Silva Rodrigues

Data da realização da defesa: 18 / dezembro / 2009

A Banca Julgadora abaixo-assinada atribuiu ao candidato o seguinte:

APROVADO

REPROVADO

São Bernardo do Campo, 18/12/2009

### MEMBROS DA BANCA JULGADORA

Prof. Dr. Paulo Sérgio Silva Rodrigues

Ass.:

Prof. Dr. Orlando Del Bianco Filho

Ass.:

Prof. Dr. Hae Yong Kim

Ass.:

### **VERSÃO FINAL DA DISSERTAÇÃO**

**ENDOSSO DO ORIENTADOR APÓS A INCLUSÃO DAS  
RECOMENDAÇÕES DA BANCA EXAMINADORA**

Aprovação do Coordenador do Programa de Pós-graduação

Prof. Dr. Marcelo Antonio Pavanello

*Aos meus pais, Herbert e Maria Luiza, à Cintia, minha querida esposa e a minha irmã, Cibele. A todos aqueles que me incentivaram a seguir os meus sonhos*

# Agradecimentos

Primeiramente, gostaria de agradecer à minha querida esposa Cintia, aos meus queridos pais, Herbert e Maria Luiza e a minha irmã, Cibele por sempre estarem ao meu lado e por sempre me amarem incondicionalmente, por sempre me apoiarem e por me incentivarem a continuar seguindo em frente.

Gostaria de agradecer também à todos os colegas do mestrado, Valquiria, Vanessa, Danilo Nunes, Danilo Eduardo, Caruso, Rodolpho Freire, Rodolfo Fussek, Thiago, Giuliano, Matheus e Rafael por terem me ajudado com dicas, esclarecimento de dúvidas, revisões, com sugestões pertinentes, e cedendo materiais e ferramentas, pelas críticas e sugestões. Aos amigos que sempre torcem por mim, essa torcida sempre me deu forças.

Não poderia nunca deixar de agradecer ao meu orientador, Paulo Sérgio Rodrigues, por ter me motivado quando necessitei, por ter me orientado quando me perdi e por ter me incentivado nos momentos difíceis. Aos Professores do mestrado, Flávio Tonindandel, Carlos Eduardo Thomaz, Paulo Eduardo Santos e Reinaldo Bianchi pela paciência e pelo empenho em passar a diante seus conhecimentos.

Agradeço também ao fisiologista Nicolae Paulescu, descobridor da insulina, graças a sua descoberta, milhões de pessoas diabéticas que antes estavam condenadas à morte, agora podem aproveitar uma vida normal da mesma forma que as pessoas saudáveis. À Daniel Peter e Henry Nestlé por terem criados juntos a melhor iguaria conhecida pelo homem redor do globo: o chocolate ao leite, doce veneno que, aliado à música Tente Outra Vez do cantor e compositor Raul Seixas me fez continuar a trabalhar nos momentos difíceis.

A ondulação deste número infinito de montanhas, cujos cumes nevados as fazem parecer como que cobertas por espuma, trouxe a minha lembrança a superfície de um oceano sendo espancado por uma tempestade. Se olhei para o oeste, o oceano exhibia-se diante de mim em toda a sua majestosa grandeza, uma continuação de certo modo, um destes lanosos topos. Onde a terra terminava e o mar começava era impossível para o olho distinguir.

*Julio Verne - Viagem ao centro da terra*

# Resumo

As interfaces gráficas ou *Graphic User Interface* (GUI) dos sistemas operacionais (SOs) modernos oferecem maneiras simplificadas para a operação das funções de um computador, ao contrário da digitação de comandos complexos em terminais de texto. Com a popularização de SOs gráficos, como por exemplo o Windows<sup>®</sup> da Microsoft<sup>®</sup>, a maioria dos programas atuais pode ser operada somente através de sua interface gráfica, gerando assim, a dificuldade de automação de tarefas por meio da interação com os componentes do programa como botões, ícones, caixas de texto, etc. Este trabalho propõe um estudo de métodos para classificação e reconhecimento de objetos sintéticos utilizando o conceito de entropia não-extensiva proposta por Constantino Tsallis de forma a possibilitar a criação de um software de automação de tarefas baseando nestas técnicas de visão computacional.

**Palavras-chave:** Objetos sintéticos. Automação de tarefas. Entropia.

# Abstract

The modern operating system's graphic user interface, or GUI, provide an easier way to access the functionalities of a computer in comparison to type long complex command strings through the keyboard on text terminals. With the spread of graphical operating systems like Windows<sup>®</sup> from Microsoft<sup>®</sup>, most of the actual programs can be used only through their graphic interface. And so, creating a problem to automate tasks interacting with the program's components like buttons, icons, textboxes, etc. In this work, it is proposed a study of sintetic image classification and sintetic object detection using the concept of non-extensive entropy proposed by Constantino Tsallis to create an automation software that can automate tasks based on computer vision techniques.

**Keywords:** sintetic objects. task automation. non-extensive entropy.

# Lista de Figuras

1.1	Execução automática - Passo 1	18
1.2	Execução automática - Passo 2	18
1.3	Execução automática - Passo 3	18
1.4	Execução automática - Passo 4	19
1.5	Execução automática - Passo 5	19
1.6	Execução automática - Passo 6	19
1.7	Imagens naturais x imagens sintéticas	23
1.8	Formulário de cadastro	23
2.1	Exemplo de captura de tela	26
2.2	Representações dos espaços	26
2.3	Histogramas	27
2.4	Cálculo automático do valor de $q$ .	36
2.5	Comparação entre os algoritmos de segmentação	38
2.6	Tipos de bordas	39
2.7	Primeira e segunda derivadas em uma linha transversal à uma borda.	39
2.8	Operadores para detecção de bordas.	40
2.9	Comparação entre os detectores de borda	40
2.10	Rede Bayesiana de um classificador	43
3.1	Exemplos de objetos	50
3.2	Etapas do programa de automação	51
3.3	Processo de detecção - Imagem original	53
3.4	Processo de detecção - Imagem segmentada	54
3.5	Processo de detecção - Filtragem de contornos	54
3.6	Processo de detecção - Objeto extraído	54
3.7	Base de Capturas	56
3.8	Base de Objetos	57
3.9	Método de Classificação	59
4.1	Resultados com limiar adaptativo	63
4.2	Detecção com entropia não-extensiva	64
4.3	Detecção com entropia não-extensiva	66
4.4	Detecção com q-entropia automatizada	67
4.5	Comparação das técnicas de detecção de objetos	69
4.6	Atributo F1 - Histograma HSV 18-3-3	70
4.7	Extração do Atributo F2	71
4.8	Atributo F2 - Variação de Intensidade Média Vertical	71
4.9	Extração do Atributo F3	72
4.10	Atributo F3 - Histograma de Raios	72
4.11	Extração do atributo F4	73
4.12	Atributo F4 - Variação Vertical de Intensidades de Cinza	73

4.13	Extração do atributo F5 . . . . .	74
4.14	Atributo F5 - Variação horizontal de de intensidades de tons de cinza . . . . .	74
4.15	Extração do atributo F6 . . . . .	75
4.16	Atributo F6 - Variação diagonal de intensidades de tons de cinza . . . . .	75
4.17	Classificação com distância euclidiana . . . . .	77
4.18	Classificação com distância clássica de Kullback-Leiber . . . . .	78
4.19	Classificação com distância não-extensiva de Kullback-Leiber . . . . .	80
4.20	Classificação com q-distância estendida de Kullback-Leiber . . . . .	81
4.21	Classificação com distância de Kullback-Leiber automatizada . . . . .	83
4.22	Comparação das técnicas de classificação . . . . .	84

# Lista de Tabelas

1.1	Seqüencia de passos utilizada no roteiro de demonstração. . . . .	17
3.1	Objetos em Interfaces Gráficas . . . . .	48
3.2	Classes de Objetos . . . . .	52
3.3	Fontes de dados . . . . .	55
3.4	Atributos extraídas dos objetos . . . . .	58
4.1	Detecção com limiar adaptativo . . . . .	63
4.2	Detecção com entropia não-extensiva . . . . .	64
4.3	Detecção com q-entropia estendida . . . . .	66
4.4	Detecção com q-entropia automatizada . . . . .	68
4.5	Comparação das técnicas de detecção de objetos . . . . .	69
4.6	Classificação com distância euclidiana . . . . .	77
4.7	Classificação com distância clássica de Kullback-Leiber . . . . .	79
4.8	Classificação com distância não-extensiva de Kullback-Leiber . . . . .	79
4.9	Classificação com q-distância estendida de Kullback-Leiber . . . . .	82
4.10	Classificação com distância de Kullback-Leiber automatizada . . . . .	83
4.11	Comparação das técnicas de classificação . . . . .	84
A.1	Detecção com q-entropia estendida . . . . .	97
B.1	Classificação com distância de Kullback-Leiber estendida . . . . .	103

# Lista de Símbolos e Abreviaturas

$HSV$	— <i>Hue, Saturation, Value</i> ou espaço de cores com três canais, matiz, saturação e valor.
$RGB$	— <i>Red, Green, Blue</i> ou espaço de cores com três canais, vermelho, verde e azul.
$2D$	— bidimensional, com duas dimensões.
$3D$	— tridimensional, com três dimensões.
$q$	— Parametro entrópico.
$\alpha$	— relação entre os parametros entrópicos $q_1$ e $q_2$ .
$H$	— histograma, distribuição de cores de uma figura.
$S_A$	— entropia para a distribuição A.
$S_q(A)$	— entropia não-extensiva para a distribuição A.
$Div_{KL}(A  B)$	— divergencia de Kullback-Leiber entre as distribuições A e B.
FP	— Falsos Positivos: Quantidade de objetos erroneamente classificados na classe em questão.
FN	— Falsos Negativos: Quantidade de objetos erroneamente classificados em classes diferentes daquela em questão.
Des.	— Desempenho percentual.
Qtd.	— Quantidade.
Bot.	— Botões.
Ico.	— Ícones.
ATx.	— Áreas de Texto.
Cxs.	— Caixas de Texto.
Obs.	— Observações.
Dist.	— Distância
KL.	— Kullback-Leiber

# Sumário

<b>LISTA DE FIGURAS</b>	<b>9</b>
<b>LISTA DE TABELAS</b>	<b>11</b>
<b>LISTA DE SÍMBOLOS E ABREVIATURAS</b>	<b>12</b>
<b>1 INTRODUÇÃO</b>	<b>16</b>
1.1 Motivação . . . . .	16
1.2 Proposta . . . . .	21
1.3 Objetos de Estudo . . . . .	22
1.4 Objetivos . . . . .	24
1.5 Exrutura do Documento . . . . .	24
<b>2 REVISÃO BIBLIOGRÁFICA</b>	<b>25</b>
2.1 Conceitos Fundamentais . . . . .	25
2.1.1 Capturas de Tela . . . . .	25
2.1.2 Representação de Imagens . . . . .	26
2.1.3 Histograma . . . . .	27
2.2 Entropia Não-Extensiva . . . . .	28
2.3 Detecção de Objetos . . . . .	31
2.3.1 Segmentação . . . . .	31
2.3.2 Detecção de Bordas . . . . .	38
2.3.3 Extração de Atributos . . . . .	41
2.4 Classificação . . . . .	42
2.4.1 Classificação com Redes Bayesianas . . . . .	42
2.4.2 Classificação com Redes Neurais . . . . .	44
2.4.3 Similaridade Entre Objetos . . . . .	44
2.5 Conclusões . . . . .	47
<b>3 PROPOSTA DE TRABALHO</b>	<b>48</b>
3.1 Objetos de Detecção . . . . .	48
3.1.1 Botões . . . . .	49
3.1.2 Ícones . . . . .	49

3.1.3	Caixas de Texto . . . . .	49
3.1.4	Áreas de Texto . . . . .	50
3.2	Visão Geral do Método Proposto . . . . .	51
3.3	Técnicas Utilizadas . . . . .	53
3.3.1	Separação do Objeto de Interesse . . . . .	53
3.3.2	Base de Dados . . . . .	55
3.3.3	Características Extraídas dos Objetos . . . . .	58
3.3.4	Classificação dos Objetos . . . . .	58
3.4	Conclusões . . . . .	60
<b>4</b>	<b>RESULTADOS OBTIDOS</b>	<b>61</b>
4.1	Metodologia Experimental . . . . .	61
4.2	Extração de Objetos . . . . .	61
4.2.1	Critério de Avaliação . . . . .	62
4.2.2	Detecção Utilizando Limiar Adaptativo . . . . .	62
4.2.3	Detecção Utilizando Entropia de Tsallis . . . . .	64
4.2.4	Detecção Utilizando Q-Entropia Estendida . . . . .	66
4.2.5	Detecção Utilizando Q-Entropia Automatizada . . . . .	67
4.2.6	Comparação dos Algoritmos de Segmentação . . . . .	69
4.3	Atributos dos Objetos . . . . .	70
4.3.1	Histograma de Cores . . . . .	70
4.3.2	Variação de Intensidade Média Vertical . . . . .	71
4.3.3	Histograma de Raios . . . . .	72
4.3.4	Variação Vertical de Intensidades . . . . .	73
4.3.5	Variação Horizontal de Intensidades . . . . .	74
4.3.6	Variação Diagonal de Intensidades . . . . .	75
4.4	Classificação de Imagens . . . . .	76
4.4.1	Critério de Avaliação . . . . .	76
4.4.2	Classificação com distância Euclidiana . . . . .	77
4.4.3	Classificação com distância de Kullback-Leiber Clássica . . . . .	78
4.4.4	Classificação com Distância de Kullback-Leiber Não-Extensiva . . . . .	79
4.4.5	Classificação com q-Distância Estendida de Kullback-Leiber . . . . .	81
4.4.6	Classificação com Distância de Kullback-Leiber Automatizada . . . . .	82
4.4.7	Comparação dos Algoritmos de Classificação . . . . .	84
<b>5</b>	<b>CONCLUSÕES FINAIS</b>	<b>86</b>
<b>6</b>	<b>TRABALHOS FUTUROS</b>	<b>88</b>
	<b>GLOSSÁRIO</b>	<b>96</b>
<b>A</b>	<b>SEGMENTAÇÃO COM Q-ENTROPIA ESTENDIDA</b>	<b>97</b>

	15
<b>B CLASSIFICAÇÃO COM DISTÂNCIA ESTENDIDA DE KULLBACK-LEIBER</b>	<b>103</b>
<b>C PARÂMETROS UTILIZADOS</b>	<b>113</b>

# Capítulo 1

## INTRODUÇÃO

Este trabalho tem como objetivo estudar a detecção de objetos de interfaces gráficas de sistemas operacionais como botões, ícones, caixas de texto, listas suspensas e grades entre outros. O estudo aqui proposto contempla todas as fases de um sistema de visão computacional, desde a detecção de objetos gráficos por meio de segmentação e a detecção de bordas, incluindo métodos de visão de alto nível como a classificação de imagens utilizando o conceito de entropia relativa como métrica de similaridade.

### 1.1 Motivação

Atualmente, a maioria dos sistemas computacionais possui uma interface gráfica que possibilita a utilização de suas funcionalidades por meio de objetos gráficos como caixas de texto, botões e ícones, para citar alguns. Tais programas permitem a interação com o usuário por meio de outros periféricos além do teclado, como o mouse ou um monitor *touchscreen*, de modo a facilitar o seu uso. Porém, estas características que buscam facilitar a utilização destes sistemas, dificultam a automação de tarefas nestes programas.

Várias das tarefas realizadas em softwares são passíveis de serem automatizadas. Entre elas se destacam o preenchimento de um formulário eletrônico por diversas vezes para o cadastro de uma grande quantidade de informações em um sistema novo. Desenvolvedores podem também desejar automatizar algumas ações para facilitar testes regressivos nas interfaces que estão desenvolvendo (Bernardo e Kon, 2008).

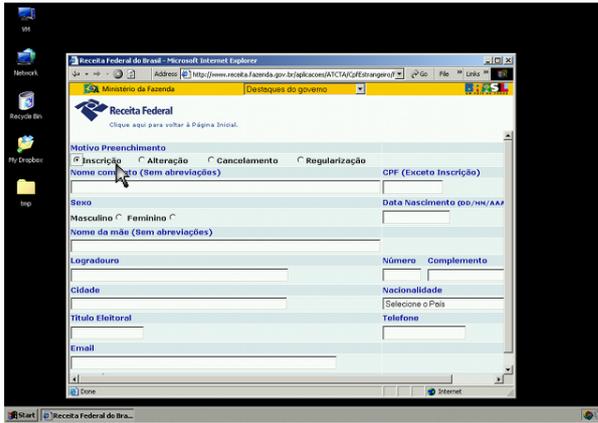
Em alguns casos específicos, quando existe acesso ao código-fonte de um programa, a automação de qualquer funcionalidade deste programa pode ser feita de maneira simples por meio da programação direta de uma rotina que execute as ações desejadas. Porém este cenário representa uma pequena parte das possíveis situações onde a automação de tarefas é necessária, uma vez que, na maioria dos casos, não se possui o código-fonte da aplicação. Nestes casos, a automação de tarefas deve ser feita por meio de uma ferramenta externa.

Existem dois tipos de ferramentas de automação que possibilitam a automação de processos em programas. A primeira, grava as interações do usuário de forma “cega”, ou seja, são gravadas as posições e cliques do mouse, além das teclas digitadas e os tempos entre cada interação. Geralmente, este tipo de automatizador deixa de funcionar se o tempo de resposta da aplicação mudar ou se os objetos mudarem de posição, fato que geralmente ocorre quando o usuário muda a janela do aplicativo de posição.

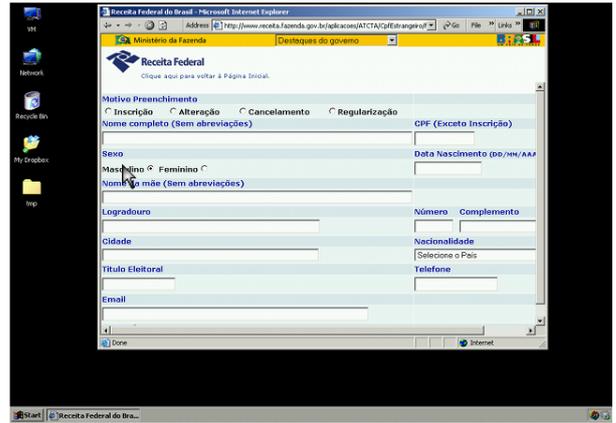
As Figuras (1.1 - 1.6) ilustram uma situação típica onde este tipo de automatizador falha. Foi criado um roteiro para preenchimento de um formulário *online* disponível no *site* da fazenda ([www.fazenda.gov.br](http://www.fazenda.gov.br)). Os passos utilizados neste roteiro estão descritos na Tabela 1.1. Na coluna (a) das Figuras (1.1 - 1.6), está demonstrada a execução de um roteiro automatizado em condições muito similares às condições de gravação, de forma que a sequência de passos é executada com perfeição. Já na coluna (b), a janela do navegador foi movida de local, prejudicando toda a execução de passos, uma vez que foram alteradas todas as coordenadas dos campos cuja interação é desejável.

Tabela 1.1: Sequência de passos utilizada no roteiro de demonstração.

<b>Passo</b>	<b>Descrição</b>
1	Seleção da opção “Inscrição”
2	Seleção da caixa de texto “Nome”
3	Digitação do nome
4	Seleção da opção “Masculino”
5	Seleção da caixa de texto “Nome da mãe”
6	Digitação do nome da mãe

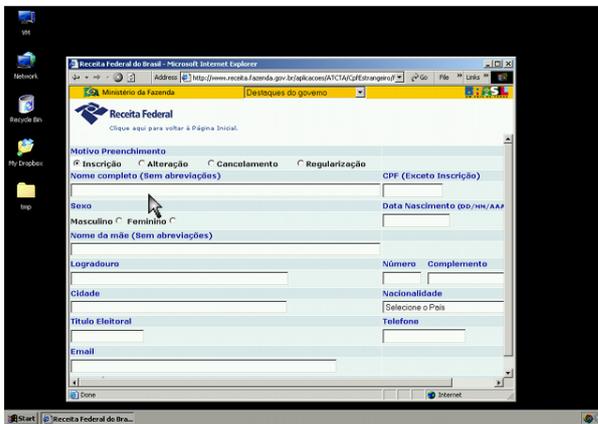


(a)

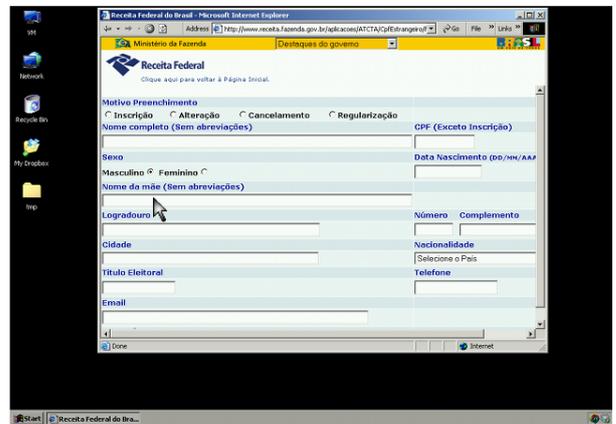


(b)

Figura 1.1: Execução de uma sequência de passos em momentos diferentes. Passo 1 - Seleciona opção “Inscrição”.

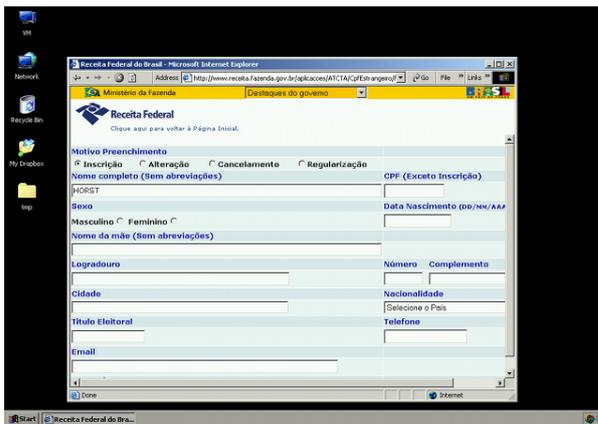


(a)

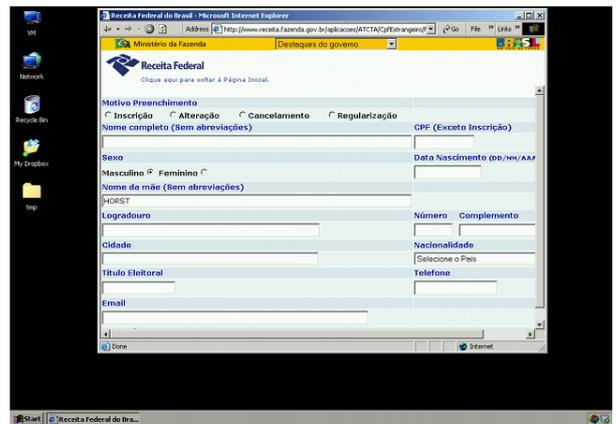


(b)

Figura 1.2: Passo 2 - Seleciona campo “Nome”.



(a)



(b)

Figura 1.3: Passo 3 - Digita nome.

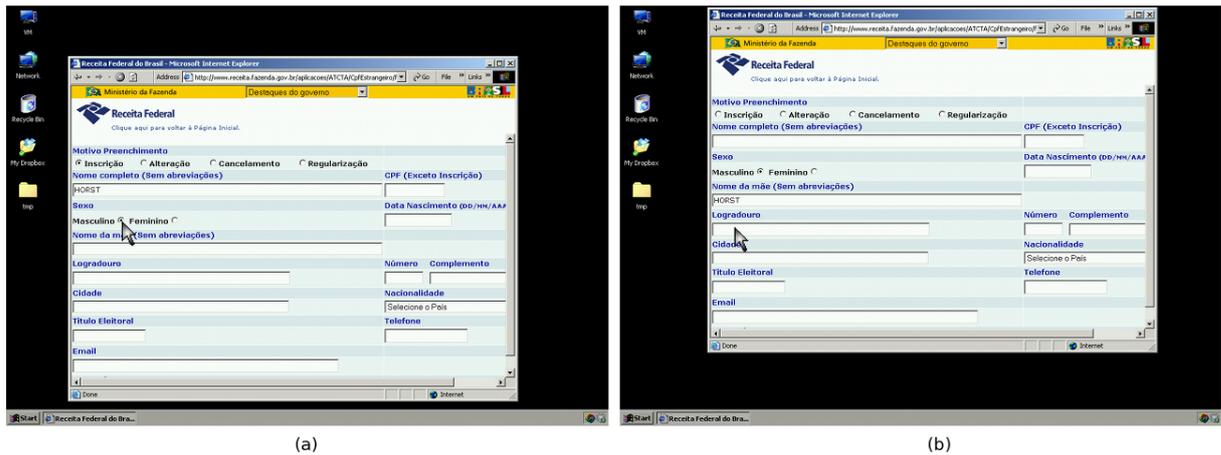


Figura 1.4: Execução de uma sequência de passos em momentos diferentes. Passo 4 - Seleciona opção “Masculino”.

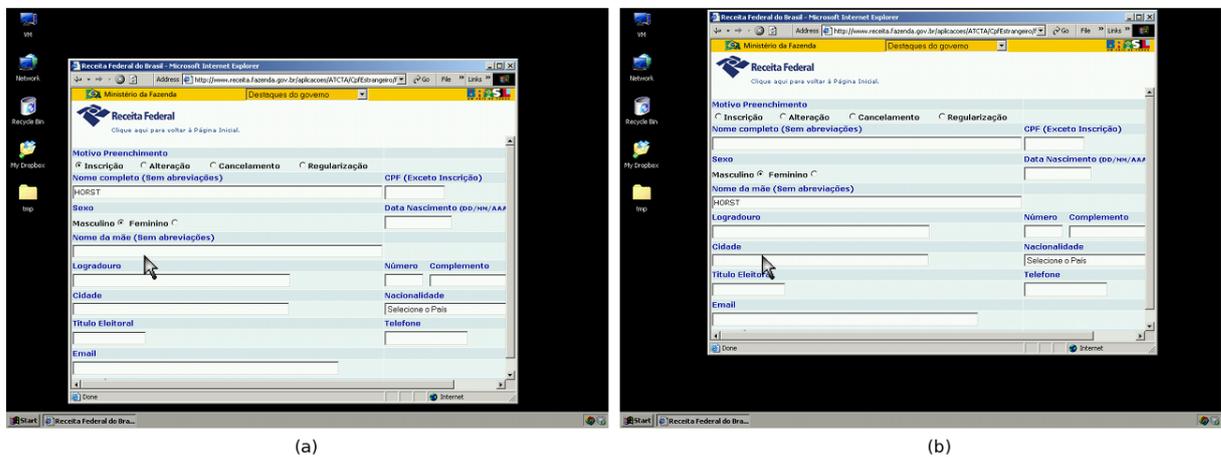


Figura 1.5: Passo 5 - Seleciona campo “Nome da Mãe”.

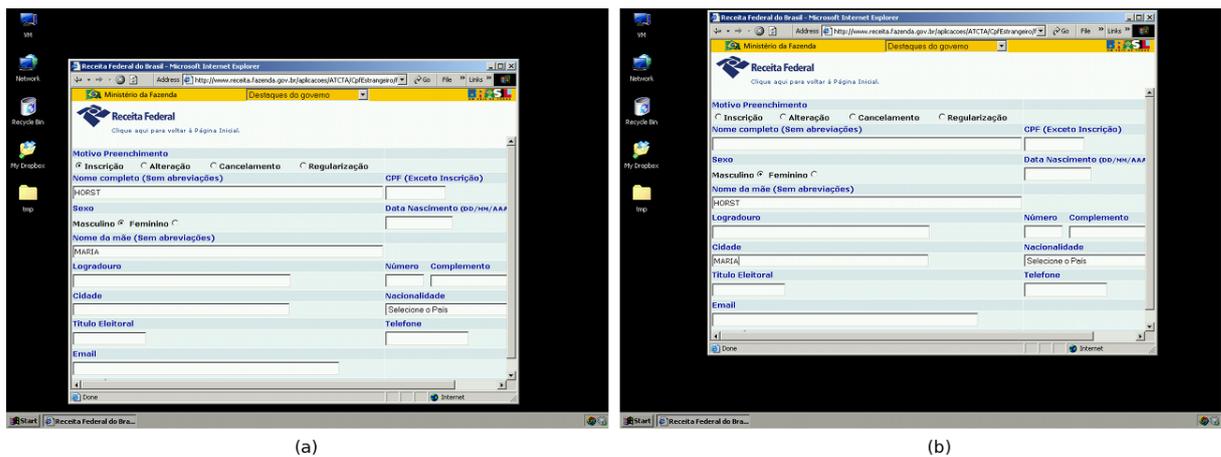


Figura 1.6: Passo 6 - Digita nome da mãe.

Por sua vez, o segundo tipo de ferramenta de automação utiliza as funcionalidades do SO para encontrar os objetos gráficos na tela e realizar as ações desejadas por meio de suas propriedades. Este apresenta um resultado melhor na detecção dos objetos em relação ao primeiro, uma vez que não depende da posição destes objetos na tela, mas sim de suas propriedades.

No entanto, este segundo tipo de automatizador é restrito a alguns sistemas operacionais proprietários. Além disto, a detecção correta dos objetos depende diretamente da linguagem em que o programa-alvo (programa cuja função se deseja automatizar) foi escrito. Por exemplo, o reconhecimento de objetos gráficos é especialmente frágil em sistemas onde os objetos são criados ou alterados dinamicamente e algumas de suas propriedades variam entre suas utilizações. Este é um cenário típico de uma aplicação *Web*. Uma vez que este tipo de programa automatizador faz uso das funcionalidades do sistema operacional para reconhecer objetos, em aplicações baseadas em acesso remoto como *Citrix*, *VNC* ou *Terminal Services* onde um servidor faz todo o processamento e retorna apenas imagens para as estações, nenhum objeto é reconhecido uma vez que todo o conteúdo exibido é uma imagem para o sistema operacional.

Geralmente, estas ferramentas de automação funcionam em duas etapas: a gravação e a reprodução. Durante a gravação, todas as interações realizadas pelo usuário são registradas bem como as propriedades dos objetos interagidos tais como botões, caixas de texto, ícones, entre outros. por meio dos dados gravados será feita a reprodução das ações do usuário.

As propriedades dos objetos utilizados são extraídas do sistema operacional por meio de seus componentes internos e de suas bibliotecas. Quando há qualquer alteração na interface do programa-alvo ou em alguma propriedade do objeto, este não é mais encontrado na fase de execução, e assim, deve-se executar uma nova gravação para que o roteiro de automação volte a funcionar.

As duas principais vantagens em automatizar um processo são a velocidade e a repetibilidade, que significa a capacidade de se reproduzir uma sequência de passos de forma exata e rápida. Sendo assim, se um processo necessita de constante manutenção em sua rotina de automação, pode-se concluir que ele tem uma baixa repetibilidade e portanto a automação deste processo pode não produzir o ganho em velocidade esperado.

Tendo em vista as limitações dos dois tipos de ferramentas de automação citadas acima, a execução “cega” do primeiro e a dependência do sistema operacional do segundo, neste trabalho propõe-se o estudo de algoritmos de visão computacional que podem ser utilizadas em um novo tipo de software de automação de processos no qual o reconhecimento de objetos em interfaces gráficas é feito por meio do uso de técnicas de visão computacional. A capacidade de detecção de objetos gráficos exclusivamente por suas características visuais permite a criação de uma nova categoria de automatizadores, muito mais robusta do que aquelas que existem atualmente, independente do sistema operacional utilizado ou da linguagem em que o programa-alvo foi escrito (Pandit e Kalbag, 1997), (Amant, 2000), (Amant e Zettlemyer, 2000), (Amant e Riedl, 2001), (Amant e Dudani, 2002).

## 1.2 Proposta

Este trabalho propõe o estudo e comparação de técnicas de detecção e classificação de objetos, utilizando principalmente algoritmos de entropia não-extensiva e analisando unicamente as propriedades visuais dos objetos como tamanho, aspecto, textura, posição, cores, etc.

Um software hipotético será utilizado como base será composto de três fases: gravação, análise e reprodução. Na etapa de gravação, as interações com o sistema operacional são gravadas bem como uma imagem da tela no momento da interação. Na fase de análise, as imagens gravadas são processadas para extração dos objetos de interesse e suas propriedades. Os objetos então são classificados para tornar sua busca mais robusta na terceira etapa. Finalmente, durante a reprodução os objetos são procurados na tela de acordo com sua classificação e, uma vez encontrados, sofrem a interação gravada na primeira fase.

Atualmente, existem inúmeras metodológicas na literatura para detecção de objetos sintéticos (Sziranyi et al., 2000), (Noyer et al., 2004), (Gouttière et al., 2007), (Rosenberger, 2006) e classificação de imagens (Santos, 2009), (Cao et al., 2009), (Ouyang et al., 2009), (Chen et al., 2009). As técnicas baseadas em entropia não-extensiva são de particular interesse desse trabalho. Essas técnicas serão utilizadas em dois estágios principais: na segmentação de capturas de tela, para definição de regiões de interesse e na classificação de imagens sintéticas, via Divergência de Kullback-Leiber para determinar a classe à qual o objeto pertence.

Esta tese se restringe aos dois principais algoritmos a serem utilizados como *engines* deste software hipotético: A detecção do objeto a partir da coordenada de interação e a classificação do objeto detectado em uma das quatro classes estudadas (Botões, Ícones, Caixas de Texto e Áreas de Texto).

### 1.3 Objetos de Estudo

O foco de estudo deste trabalho são objetos sintéticos gerados de forma artificial por programas com interface gráfica. Como exemplo destes tipos de objetos pode-se citar botões, ícones, caixas de texto, caixas de seleção, entre outros. Um sistema de processamento de imagens sintéticas possui diversas etapas e características em comum com os sistemas de processamento de imagens naturais: Em ambos os casos, as imagens variam em forma, cor e posição ou ainda podem sofrer oclusão parcial ou total.

O reconhecimento dos objetos sintéticos é um problema interessante e desafiador de ser estudado, uma vez que o objeto identificado no instante  $t_0$  poderá sofrer alterações de posição, cor, forma, textura ou ainda de *aliasing* (suavização das diferenças de intensidades entre pixels vizinhos) entre o instante  $t_0$  e o instante da reprodução  $t_1$ . Essas alterações podem ser provenientes de atualizações ou customizações do sistema operacional bem como atualizações do programa-alvo. Atualmente é comum programas sofrerem alterações em sua interface quando uma nova versão do *software* é lançada.

Além desses fatos, os objetos podem sofrer oclusão parcial ou completa, dificultando seu reconhecimento. Existem ainda outros obstáculos como a descontinuidade de objetos, a transparência de objetos nos sistemas operacionais mais modernos e a alteração dos objetos que ocorre quando o mouse passa sobre eles. A Figura 1.7 mostra um exemplo de algumas variações de objetos sintéticos e naturais. Assim como objetos em imagens naturais, os objetos sintéticos também sofrem alterações de cor, forma e textura. A existência de objetos similares na tela constitui outro obstáculo à detecção do objeto correto, visto que, nestes casos, para um mesmo objeto, existirão vários candidatos com a mesma medida de distância. Por exemplo, em um mesmo instante, podem existir na tela diversas caixas de texto exatamente iguais, uma para cada dado, como pode-se observar na Figura 1.8.

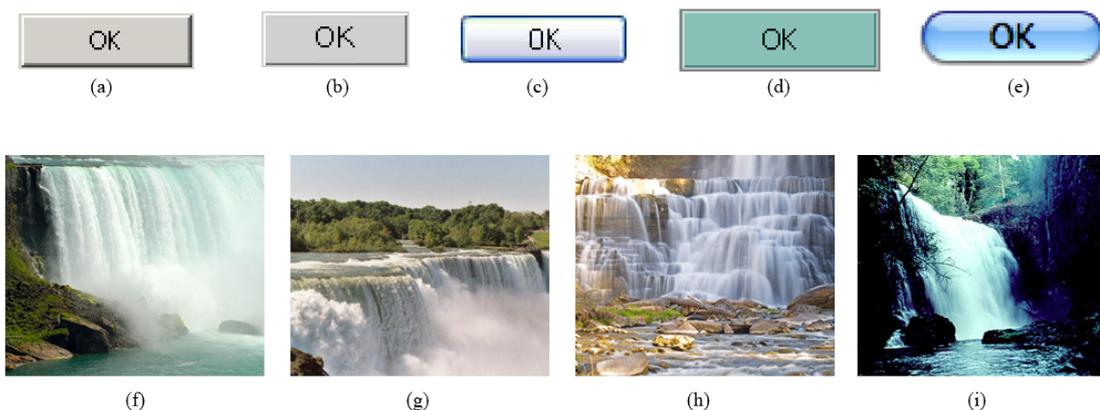


Figura 1.7: Exemplo de variações de imagens naturais e de objetos sintéticos. As imagens naturais, extraídas da internet, (f),(g),(h) e (i) representam quedas d’água enquanto que as imagens sintéticas, extraídas da base de imagens utilizada neste trabalho,(a),(b),(c),(d) e (e) representam um botão.

Ministério da Fazenda Destaque do governo

**Receita Federal**  
Clique aqui para voltar à Página Inicial.

<b>Motivo Preenchimento</b>	
<input type="radio"/> Inscrição <input type="radio"/> Alteração <input type="radio"/> Cancelamento <input type="radio"/> Regularização	
<b>Nome completo (Sem abreviações)</b>	<b>CPF (Exceto Inscrição)</b>
<input type="text"/>	<input type="text"/>
<b>Sexo</b>	<b>Data Nascimento (DD/MM/AAAA)</b>
Masculino <input type="radio"/> Feminino <input type="radio"/>	<input type="text"/>
<b>Nome da mãe (Sem abreviações)</b>	
<input type="text"/>	
<b>Logradouro</b>	<b>Número    Complemento</b>
<input type="text"/>	<input type="text"/> <input type="text"/>
<b>Cidade</b>	<b>Nacionalidade</b>
<input type="text"/>	Selecione o País <input type="text"/>
<b>Título Eleitoral</b>	<b>Telefone</b>
<input type="text"/>	<input type="text"/>
<b>Email</b>	
<input type="text"/>	
<input type="button" value="Enviar"/>	

Figura 1.8: Formulário extraído da página da Receita Federal na internet. Os campos “Nome completo” e “Nome da Mãe” possuem as mesmas características visuais.

## 1.4 Objetivos

Os objetivos deste trabalho são: estudar as características de objetos gráficos de um sistema operacional, estudar e comparar de técnicas de detecção e classificação de objetos, utilizando principalmente algoritmos de entropia não-extensiva de modo a detectá-los e classificar-los em diferentes cenários com robustez em relação a pequenas variações de localização, tamanho, cor, forma e textura.

## 1.5 Estrutura do Documento

Esta tese está dividida em mais cinco capítulos. No próximo, serão vistos conceitos fundamentais para o entendimento do método proposto e menções a trabalhos relacionados a este por meio de métodos utilizados na solução de problemas similares. No terceiro capítulo está descrita a proposta para a detecção e classificação dos objetos de interesse. O quarto capítulo discute os resultados obtidos dos testes executados cujas conclusões estão no capítulo cinco. Finalmente, o último capítulo descreve possíveis extensões deste trabalho.

## Capítulo 2

# REVISÃO BIBLIOGRÁFICA

Este capítulo descreve algumas das técnicas de visão computacional que podem ser utilizadas para detecção e classificação de objetos. Inicialmente serão vistos conceitos fundamentais, seguidos das técnicas de visão de baixo e médio nível para extração do objeto de interesse e das técnicas de visão de alto nível para classificação.

### 2.1 Conceitos Fundamentais

Os objetos de estudo deste trabalho são imagens sintéticas geradas por um sistema operacional. Portanto, serão utilizadas imagens instantâneas da tela nos momentos de interação com o usuário como eventos de cliques do mouse ou digitação no teclado.

#### 2.1.1 Capturas de Tela

Tais imagens, conhecidas como *screenshots* ou capturas de tela, são retiradas da memória de vídeo do computador e assim não possuem ruído e tampouco variação de iluminação. As capturas de tela utilizadas neste trabalho possuem três principais características: a resolução máxima de 800 *pixels* de largura por 600 *pixels* de altura, a cobertura de toda a área da trabalho disponível e não apenas parte dela; e uma imagem de fundo que não se mescle às janelas abertas na área de trabalho (geralmente fundo preto ou azul), uma vez que, de outra forma, a imagem de fundo pode possuir figuras que sejam semelhantes aos objetos procurados e, assim, confundir o algoritmo de detecção. A Figura 2.1 mostra um exemplo dos *screenshots* utilizados.

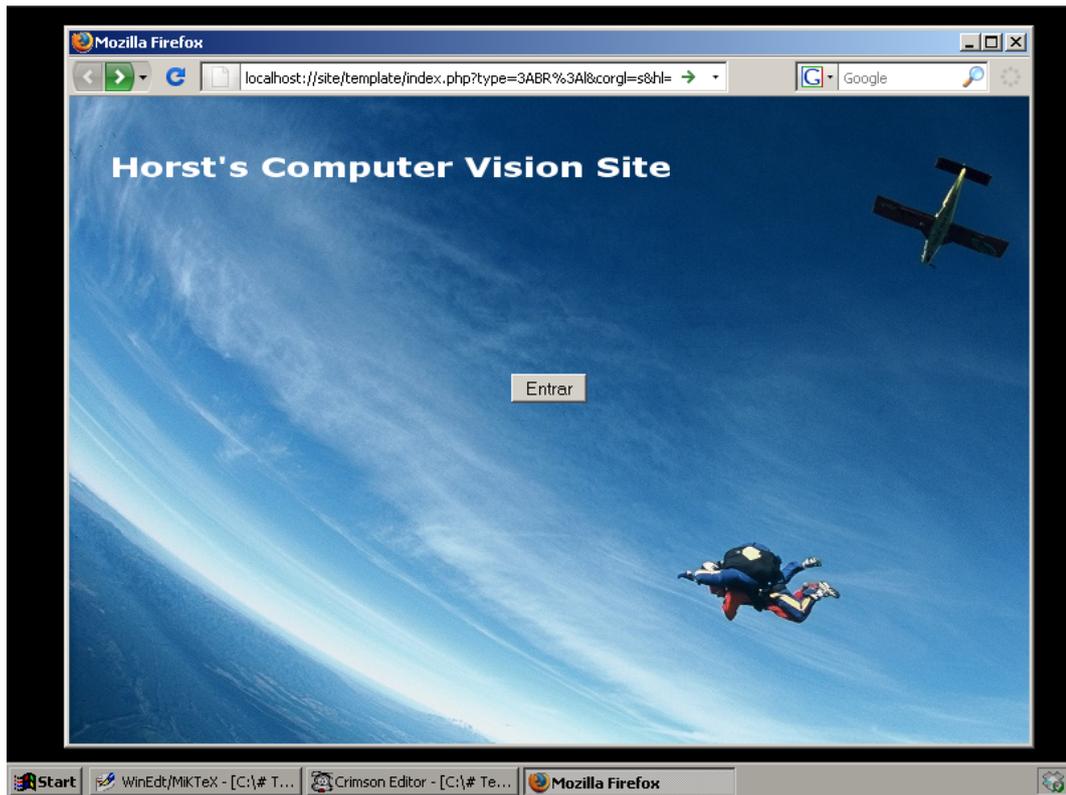


Figura 2.1: Exemplo de uma das capturas de tela utilizadas, com as três características necessárias: 1 - Resolução de 800x600; 2 - Cobertura de toda a área disponível, e não somente a janela ativa; 3 - Fundo preto

## 2.1.2 Representação de Imagens

Uma imagem pode ser representada de várias maneiras diferentes, O espaço de representação de cores mais comum em imagens digitais é o RGB, que consiste de três canais, um para cada cor primária. Nesta espaço, cada pixel é representado por uma tripla de valores ( $R$ ,  $G$ ,  $B$ ). Um outro espaço de representação de cores é o HSV. Criado em 1978 por Alvy Ray Smith, este modelo representa uma imagem em três canais: a matiz ( $H$ ), a saturação ( $S$ ) e o valor ( $V$ ). A Figura 2.2 mostra uma representação das cores nos espaço RGB e HSV (Gonzalez e Woods, 2001).

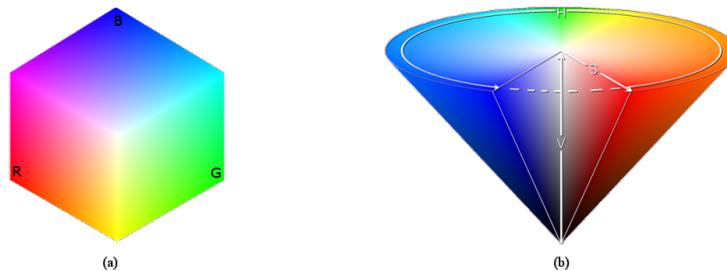


Figura 2.2: Representações dos espaços RGB em (a) e HSV em (b).

### 2.1.3 Histograma

A distribuição de cores dos pixels varia de imagem para imagem. A forma mais comum de quantificar as cores de uma figura é o seu histograma, que é um mapa com uma contagem de pixels para cada intensidade de cor que uma imagem pode ter. O histograma de uma imagem digital é uma função discreta dada pela Equação (2.1) (Gonzalez e Woods, 2001). A Figura 2.3 demonstra um exemplo de três histogramas, um de cada canal de cor, extraídos de uma imagem natural.

$$h(r_k) = \frac{n_k}{n} \quad (2.1)$$

onde:  $r_k$  é a intensidade de cor  $k$ ;  $h(r_k)$  é o valor normalizado do histograma para  $r_k$ ;  $n_k$  é a quantidade de pixels com intensidade  $r_k$  e  $n$  é a quantidade total de pixels na imagem.

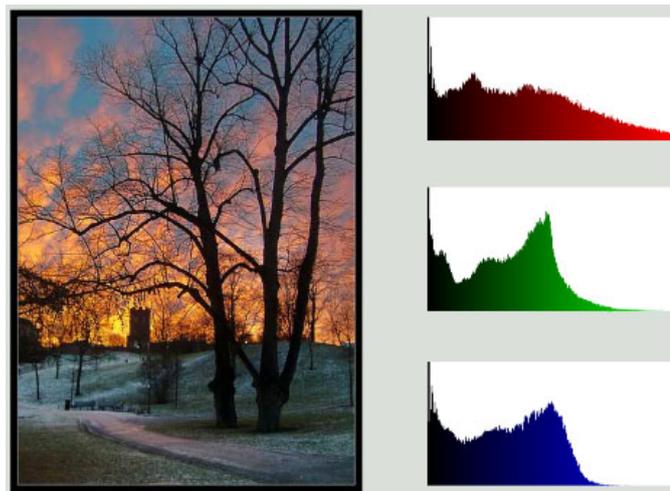


Figura 2.3: Histogramas dos canais vermelho, azul e verde de uma imagem natural.

Sendo um histograma uma distribuição de probabilidades, qualquer distribuição de valores de um contexto comum pode ser chamada de histograma e, sendo assim, podemos ter vários histogramas diferentes para uma mesma imagem: histograma de tons de cinza, das variações HSV, de intensidades ao longo dos eixos horizontal e vertical ou ainda histograma de raios no caso de análise de imagens de diferentes tamanhos. Neste trabalho, uma vez que as informações de cor são fundamentais para a metodologia proposta, o histograma de cores de um objeto é calculado com base no espaço HSV, limitado a dezoito tons de matiz, três de saturação e três de intensidade, gerando um vetor de características de 162 entradas, o que garante um baixo custo computacional para manipulá-lo (Rodrigues, 2003).

## 2.2 Entropia Não-Extensiva

A entropia foi utilizada inicialmente na área da Termodinâmica no século XIX Por Rudolf Clausius, que deu o nome de entropia para uma troca irreversível de calor (Clausius, 1879). Sua definição permitiu a estruturação da teoria da Termodinâmica. Em 1877, Ludwig Boltzmann formulou a expressão de entropia descrita na Equação (2.2) (Boltzmann, 1872).

$$S_B = k_B \cdot \ln(\omega) \quad (2.2)$$

Onde:  $S_B$  é a entropia de Boltzmann;  $k_B$  é a constante de Boltzmann e  $\omega$  é o número de microestados do sistema.

No ano seguinte, 1873, William Gibbs estendeu a expressão de Boltzmann como pode-se observar na Equação (2.3), tornando-a mais abrangente, e assim, iniciando a formulação da Mecânica Estatística (Gibbs, 1873), tornando a equação (2.2) conhecida como fórmula da entropia de Boltzmann-Gibbs.

$$S_G = -k_B \sum_{i=1}^{\omega} p_i \cdot \ln(p_i) \quad (2.3)$$

onde:  $S_G$  é a entropia de Gibbs;  $k_B$  é a constante de Boltzmann;  $\omega$  é o número de microestados do sistema e  $p_i$  é a probabilidade do sistema estar no microestado  $i$ .

No século seguinte, em 1948, Claude Shannon associou a entropia à quantidade de informação ao tentar maximizar a quantidade de informação que podia ser transferida em um sistema telefônico (Shannon, 1948). Para tanto, ele definiu como quantidade de informação como a dificuldade de previsão de uma mensagem, ou seja, a quantidade de informação está relacionada com a possibilidade da mesma ocorrer. Uma mensagem que ocorre com muita frequência possui pouca informação, enquanto que uma mensagem que ocorre raramente possui muita informação (Tavares, 2003). Shannon então aplicou o conceito de entropia da Mecânica Estatística de Gibbs em sua medida de quantidade de informação como pode ser observada na Equação (2.4).

$$H = -k_S \sum_{i=1}^n p_i \cdot \ln(p_i) \quad (2.4)$$

onde:  $k_S$  é uma constante positiva;  $n$  é o número de eventos e  $p_i$  é a chance do evento  $i$  ocorrer.

A entropia de Boltzmann-Gibbs, também conhecida por entropia clássica, tem como propriedade a regra da adição, de forma que, para duas distribuições independentes  $P_A = [p_1, \dots, p_n]$  e  $P_B = [q_1, \dots, q_m]$ , a entropia da distribuição composta  $(P_A * P_B) = \{p_i q_j\}_{i,j}$  onde  $1 \leq i \leq n$  e  $1 \leq j \leq m$  é igual a soma das entropias de  $P_A$  e  $P_B$  (Tavares, 2003):

$$H(P_A * P_B) = H(P_A) + H(P_B) \quad (2.5)$$

As técnicas baseadas na entropia tradicional de Boltzmann-Gibbs, como por exemplo (Bolzan et al., 2004) (Martínez-Finkelshtein e Sánchez-Lara, 2007), (Coppi e D'Urso, 2006), (Piera e Parada, 2009) assumem que não existem interações de longo alcance entre as características dos sistemas. No entanto, sabe-se, a partir de trabalhos como (Taruya e Sakagami, 2002) (Beck, 2002) (Martin et al., 2004), que podem haver interações de longo alcance tanto espaciais quanto temporais. Ou seja, interações que podem fazer com que o sistema de estudo tenha algumas de suas propriedades alteradas. No caso de imagens, uma alteração de uma característica pode fazer com que um objeto possa ser classificado de forma diferente, ou com que o sistema que o está processando tome ações diferentes, de acordo com a alteração destas propriedades. Em sistemas onde estes tipos de interações ocorrem, a entropia tradicional de Shannon não é suficiente para medir o sistema físico composto pelos objetos e suas características, necessitando de uma extensão para poder ser capaz de atender a sistemas físicos mais completos.

Em 1988, Constantino Tsallis propôs uma generalização da entropia de Boltzmann-Gibbs para atender a sistemas não-extensivos, cuja forma mais conhecida é dada pela Equação (2.6) (Tsallis, 1988). Esta técnica também é conhecida por entropia não-extensiva, entropia de Tsallis ou ainda por  $q$ -entropia, devido ao parâmetro  $q$  em sua fórmula (Esquef, 2002).

$$S_q(p) = \frac{1}{q-1} \left( 1 - \sum_x p^q(x) \right) \quad (2.6)$$

onde:  $q$  é o parâmetro entrópico e  $p$  é a distribuição probabilística sobre a variável  $x$ .

No limite  $q \rightarrow 1$ , a Equação (2.6) se reduz a entropia de Boltzmann-Gibbs descrita na Equação (2.3). O novo formalismo proposto por Tsallis, embora tenha sido originalmente proposta para aplicações diretas na mecânica estatística, assim como ocorreu com a entropia de Shannon, pode ser também imediatamente aplicado às diversas áreas do conhecimento, possibilitando o uso da entropia não-extensiva em campos onde a forma tradicional não se adequava, dentre os quais

pode-se citar o movimento das Galáxias (Nakamichi e Morikawa, 2004), análise de DNA (Karmeshu e Krishnamachari, 2004), estudo de raios cósmicos, lingüística, impulsos cerebrais (Tong et al., 2003), medicina, engenharia, lingüística, matemática (Tadaki, 2008), (Tavares, 2003), (Esquef, 2002) e também no campo de visão computacional (Rodrigues et al., 2005), (Zhang e Wu, 2008), (Rodrigues e Giraldi, 2009b).

Em 2004, Albuquerque *et. al* utilizaram o conceito de entropia não-extensiva de Tsallis para segmentação de imagens de mamografia (Albuquerque et al., 2004). Para tanto, os autores utilizaram o histograma de tons de cinza para gerar duas distribuições probabilísticas, uma para o *foreground*, e outra para o *background* de forma que o ponto de separação entre elas é definido pela entropia máxima de cada distribuição. Esta idéia foi utilizada e estendida por Rodrigues e Giraldi nos anos seguintes (Rodrigues et al., 2005), (Rodrigues et al., 2006a), (Rodrigues et al., 2006b), (Rodrigues et al., 2006d), (Rodrigues et al., 2006c), (Rodrigues e Giraldi, 2009b), (Rodrigues e Giraldi, 2009a).

O valor do parâmetro entrópico  $q$  que propicia o resultado mais adequado depende diretamente do problema e geralmente é obtido experimentalmente após algumas execuções do algoritmo com valores diferentes de  $q$ . Além disso, é possível demonstrar que a entropia de Tsallis não é aditiva, uma vez que, para duas distribuições aleatórias independentes  $A = [p_1, \dots, p_n]$  e  $B = [q_1, \dots, q_m]$ , a entropia da distribuição composta ( $A * B$ ) não é igual a soma das entropias de  $A$  e  $B$ , diferentemente da entropia clássica. Assim, a esta propriedade dá-se o nome de pseudo-aditividade (Shannon e Weaver, 1948), (Tsallis, 2001) (Albuquerque et al., 2004), (Rodrigues e Giraldi, 2009b):

$$S_q(A * B) = S_q(A) + S_q(B) + (1 - q) \cdot S_q(A) \cdot S_q(B) \quad (2.7)$$

Esta propriedade nos fornece uma medida de informação da distribuição composta ( $A * B$ ) e é de fundamental importância para a segmentação de imagem baseada em entropia como será visto mais a frente na Seção 2.3.1.

## 2.3 Detecção de Objetos

Uma captura de tela é uma imagem e possui muitas informações sobre os objetos contidos nela, porém estes dados necessitam ser obtidos através de técnicas de processamento de imagens digitais tais como a segmentação, a criação do mapa de bordas e finalmente a classificação dos objetos, que são passos fundamentais para extração de objetos em um sistema de processamento de imagens digitais (Esquef, 2002).

### 2.3.1 Segmentação

Dentro de uma imagem digital, seja ela natural ou sintética, cada objeto possui características distintas de cor, forma e textura. O processo de segmentação é a primeira etapa onde estes objetos são separados. De acordo com (Forsyth e Ponce, 2002), nesta etapa, os pixels de uma figura são agrupados em regiões da imagem conhecidas como *clusters de pixels*. Quando a imagem é separada em apenas dois agrupamentos, pode-se chamar este processo de binarização.

Ainda não foi descoberta uma forma definitiva de segmentação capaz de separar com exatidão todos os objetos em uma imagem (Jain, 2009). Por este motivo, existem diversos métodos diferentes que podem ser utilizados para a realização da segmentação de uma imagem, cada um com suas peculiaridades. Dentre os algoritmos de segmentação mais conhecidos estão o  $k$ -means (Jain, 2009), a segmentação através de entropia (Albuquerque et al., 2004) e o limiar adaptativo (Sezgin e Sankur, 2004).

O método de segmentação mais utilizado é aquele através de limiar ou *threshold*. Devido à sua simplicidade, é também o algoritmo de segmentação mais veloz, com complexidade  $O(n)$ , uma vez que cada pixel é acessado somente uma vez. Este algoritmo faz uso de uma função degrau com um valor fixo de comparação  $t$ . Os pixels com maior intensidade recebem um valor máximo e aqueles com menor intensidade recebem o valor mínimo. Em sua forma adaptativa, o valor de comparação varia de acordo com a região da imagem (Gonzalez e Woods, 2001), (Shapiro e Stockman, 2000). Este algoritmo faz parte de uma das técnicas utilizadas por Pi e Zhang para segmentar imagens com regiões homogêneas e com granularidades (Pi e Zhang, 2005). Também foi utilizada em muitos outros trabalhos para binarizar imagens (Blayvas et al., 2006), (Sezgin e Sankur, 2004), (Sang et al., 2007), (Pham et al., 2007), (Pereira, 2009).

A definição de  $t$  pode ser feita manualmente, de forma adaptativa, ou ainda utilizando entropia não-extensiva (Albuquerque et al., 2004). Nesta última forma, o histograma  $H$  de cores é dividido em duas partes  $H_A$  e  $H_B$ , com o ponto de separação variando do início ao fim do histograma. Para cada posição de  $H$ , são calculadas as entropias das duas partes do Histograma  $S_{H_A}$  e  $S_{H_B}$ . Então, utilizando-se a Equação (2.7), descobre-se o ponto de comparação  $t_{opt}$  ótimo, no qual a entropia conjunta é máxima, como pode-se observar no Algoritmo (1).

---

**Algoritmo 1** Segmentação com Entropia Não-extensiva
 

---

$H$  = histograma da imagem

$L$  = posições do histograma

$S_{Max} = -\infty$

$t_{opt}=1$

**PARA** ( $t=1$  até  $L$ ) **FAZER**

$A = H[1 \dots t]$

$B = H[t + 1 \dots L]$

normaliza( $A$ )

normaliza( $B$ )

$S_A$  = calcular a q-Entropia de  $S_A$  segundo Equação (2.6)

$S_B$  = calcular a q-Entropia de  $S_B$  segundo Equação (2.6)

$S_T$  = calcular a entropia conjunta de ( $A * B$ ) segundo Equação (2.7)

**SE**  $S_T > S_{Max}$  **ENTÃO**

$S_{Max} = S_T$

$t_{opt} = t$

**FIM SE**

**FIM PARA**

---

Em 2009, Rodrigues e Giraldo propuseram uma extensão para a equação de Tsallis adicionando mais um parâmetro entrópico na Equação (2.6), conforme prova a seguir. Pode-se dizer que para uma distribuição contínua aleatória  $X$  e sua função de distribuição de probabilidades  $p(X)$ , a entropia de Tsallis é definida por (Rodrigues e Giraldo, 2009a):

$$S_q = \frac{1 - \int_{-\infty}^{+\infty} [p(x)]^q dx}{q - 1}. \quad (2.8)$$

Conseqüentemente, as entropias do *foreground*  $S_q(A)$  e do *background*  $S_q(B)$  podem ser definidas da seguinte forma:

$$S_q(A) = \frac{1 - \int_{-\infty}^t [p_A(x)]^q dx}{q - 1}, \quad (2.9)$$

$$S_q(B) = \frac{1 - \int_{-\infty}^t [p_B(x)]^q dx}{q - 1}. \quad (2.10)$$

Estas equações são válidas nas seguintes condições:

$$\int_{-\infty}^t p_A(x) dx = 1, \quad (2.11)$$

$$\int_t^{+\infty} p_B(x) dx = 1. \quad (2.12)$$

$$p_B(x) \geq 0, \quad p_A(x) \geq 0, \quad \forall x \in \mathfrak{R}. \quad (2.13)$$

Então, substituindo as Expressões (2.9) e (2.10) na Equação (2.7), obtém-se a medida de informação entre o *foreground* e o *background*. Assim, é possível desenvolver uma formulação contínua para o problema de otimização definido na Expressão (2.7). Uma condição necessária para maximizar  $S_q(A * B)$  é  $dS_q(A * B)/dt = 0$ , que pode ser decomposto utilizando álgebra simples:

$$(-[p_A(t)]^q + [p_B(t)]^q) + (1 - q)(-[p_A(t)]^q S_q(B) + [p_B(t)]^q S_q(A)) = 0. \quad (2.14)$$

A Equação (2.14) pode ser reescrita da seguinte maneira:

$$\left(-\left[\frac{p(t)}{\Lambda_A}\right]^q + \left[\frac{p(t)}{\Lambda_B}\right]^q\right) + (1 - q)\left(-\left[\frac{p(t)}{\Lambda_A}\right]^q S_q(B) + \left[\frac{p(t)}{\Lambda_B}\right]^q S_q(A)\right) = 0. \quad (2.15)$$

onde  $\Lambda_A$  e  $\Lambda_B$  são as áreas sobre as funções  $p(x)$ , antes e depois da solução  $t = t_{opt}$ , respectivamente.

Uma solução independente de  $q$  para esta expressão é  $S_q(A) = S_q(B) = S$  e  $\Lambda_A = \Lambda_B$ . A partir das expressões (2.9) e (2.10), nota-se que:

$$\begin{aligned} S_q(A) = S_q(B) &\Leftrightarrow \int_{-\infty}^t \left[\frac{p(t)}{\Lambda_A}\right]^q dx = \int_t^{+\infty} \left[\frac{p(t)}{\Lambda_B}\right]^q dx \Leftrightarrow \\ &\int_{-\infty}^t \frac{[p(x)]^q}{\Lambda_A^q} dx = \int_t^{+\infty} \frac{[p(x)]^q}{\Lambda_B^q} dx, \\ \frac{\Lambda_A^q}{\Lambda_B^q} &= \frac{\int_{-\infty}^t [p(x)]^q dx}{\int_t^{+\infty} [p(x)]^q dx}. \end{aligned} \quad (2.16)$$

A Expressão (2.16) será utilizada como heurística para o método proposto em seguida. As Equações (2.15) e (2.16) podem ser utilizadas em uma solução independente de  $q$  para maximização de  $S_q(A * B)$ . No entanto, esta forma tem um custo computacional alto, além do fato de que o valor  $t$  pode não existir. Além disto, geralmente a solução é um valor de  $t_{opt}$  que pode não ser o

mais eficiente para segmentação. Sendo assim, Rodrigues e Giraldi propuseram um método para evitar tais problemas (Rodrigues e Giraldi, 2009a), redefinindo as métricas  $S_q(A * B)$  como:

$$S_{q_A, q_B}(A * B) = S_{q_A}(A) + S_{q_B}(B) + \left(1 - \left(\frac{q_A + q_B}{2}\right)\right) S_{q_A}(A) S_{q_B}(B) \quad (2.17)$$

onde  $q_A$  e  $q_B$  são dois valores independentes do parâmetro entrópico. Assim, uma generalização da Equação (2.14) é dada por:

$$\left(\frac{-[p_A(t)]^{q_A}}{q_A - 1} + \frac{[p_B(t)]^{q_B}}{q_B - 1}\right) + \left(1 - \left(\frac{q_A + q_B}{2}\right)\right) \left(\frac{-[p_A(t)]^{q_A}}{q_A - 1} S_{q_B}(B) + \frac{[p_B(t)]^{q_B}}{q_B - 1} S_{q_A}(A)\right) = 0. \quad (2.18)$$

A partir desta expressão, observa-se que  $S_{q_A}(A) = S_{q_B}(B) = S$  e  $p_A(t) = p_B(t)$  não é uma solução direta se  $q_A \neq q_B$ . Dessa forma, a solução para o problema de otimização não está limitada à  $S_{q_A}(A) = S_{q_B}(B) = S$ , o que pode introduzir mais um grau de liberdade no algoritmo de segmentação com q-entropia, se  $S_q(A * B)$  for substituído por  $S_{q_A, q_B}(A * B)$ . Em contrapartida, deve-se colocar dois valores para o parâmetro entrópico  $q$ . Para atender a este problema, retorna-se à Equação (2.16) e se propõe a generalização abaixo:

$$\frac{\Lambda_A^{q_A}}{\Lambda_B^{q_B}} = \frac{\int_{-\infty}^t [p(x)]^{q_A} dx}{\int_t^{+\infty} [p(x)]^{q_B} dx}. \quad (2.19)$$

Para simplificar esta equação, é possível considerar algumas heurísticas com o objetivo de obter de uma expressão envolvendo a razão  $q_A/q_B$ . Supondo que:

$$\frac{\int_{-\infty}^t [p(x)]^{q_A} dx}{\int_t^{+\infty} [p(x)]^{q_B} dx} = \alpha^{q_B}, \quad (2.20)$$

onde  $\alpha$  é uma constante. Substituindo a Equação (2.20) na Expressão (2.19) e tomando-se o log dos dois lados, tem-se:

$$\frac{q_B}{q_A} = \frac{\ln \Lambda_A}{\ln(\alpha \Lambda_B)} = \frac{\ln \int_{-\infty}^t [p(x)] dx}{\ln(\alpha \int_t^{+\infty} [p(x)] dx)} \quad (2.21)$$

Então, a alteração proposta pelos autores na entropia de Tsallis é dada por (Rodrigues e Giraldi, 2009a):

$$S_{q_A, q_B}(A * B)(t) = \frac{1 - \sum_{i=1}^t (p_A)^{q_A}}{q_A - 1} + \frac{1 - \sum_{i=t+1}^k (p_B)^{q_B}}{q_B - 1} + \left(1 - \frac{q_A + q_B}{2}\right) \cdot \frac{1 - \sum_{i=1}^t (p_A)^{q_A}}{q_A - 1} \cdot \frac{1 - \sum_{i=t+1}^k (p_B)^{q_B}}{q_B - 1} \quad (2.22)$$

Nota-se que, para  $q_A = q_B = q$ , a Equação (2.22) volta a forma de Tsallis dada pela Equação (2.6). Iniciando o algoritmo com  $q = q_A$ , pode-se calcular o valor correspondente de  $q_B$  por meio da Equação (2.21), e assim, resolver o problema de maximização definido na Equação (2.7) com  $S_q(A * B)$  substituído por  $S_{q_A, q_B}(A * B)$ . Finalmente, o algoritmo para segmentação utilizando a extensão da entropia não-extensiva de Tsallis proposta por Rodrigues e Giraldi é dado por:

---

**Algoritmo 2** Segmentação com Entropia Não-Extensiva Estendida

---

Calcular  $topt$  para  $q_A$  utilizando o Algoritmo (1);

Calcular  $q_B$  utilizando  $topt$  e  $q_A$  por meio da Equação (2.21);

Segmentar novamente com o Algoritmo (1), utilizando  $S_{q_A, q_B}(A * B)$ ;

---

Um problema que ocorre na utilização da entropia de Tsallis, no algoritmo proposto por Albuquerque *et al* (Albuquerque et al., 2004), e também em sua versão estendida (Rodrigues e Giraldi, 2009a), é a definição do valor de  $q$ . Assim sendo, Rodrigues e Giraldi, propuseram um método para cálculo automático do valor de  $q$  (Rodrigues e Giraldi, 2009b). Dado que a entropia máxima de uma distribuição  $X$  ocorre quando todos os estados de  $X$ ,  $(x_1, x_2, \dots, x_n)$  possuem a mesma probabilidade, então a entropia máxima de uma distribuição  $X$ ,  $S_{MAX}$  é dada pela Equação (2.23).

$$S_{MAX} = \frac{1}{q - 1} [1 - n(p^q(x))] \quad (2.23)$$

onde:  $q$  é o parâmetro entrópico e  $n$  é o total de elementos da distribuição  $X$ .

Segundo os autores Rodrigues e Giraldi, do ponto de vista da Teoria da Informação, quanto menor for a relação entre a entropia  $S_q$  produzida por um valor  $q$  e a entropia máxima  $S_{MAX}$  de um sistema, maior é a informação contida no sistema. Este é um princípio bem conhecido da Teoria da Informação e dá a idéia de que o valor ótimo de  $q$  pode ser calculado minimizando a função  $S_q/S_{MAX}$  (Rodrigues e Giraldi, 2009b). Dessa forma, para cada imagem, é calculada a relação entre a entropia  $S_q$  e a entropia máxima  $S_{MAX}$  variando-se o valor de  $q$  na faixa de

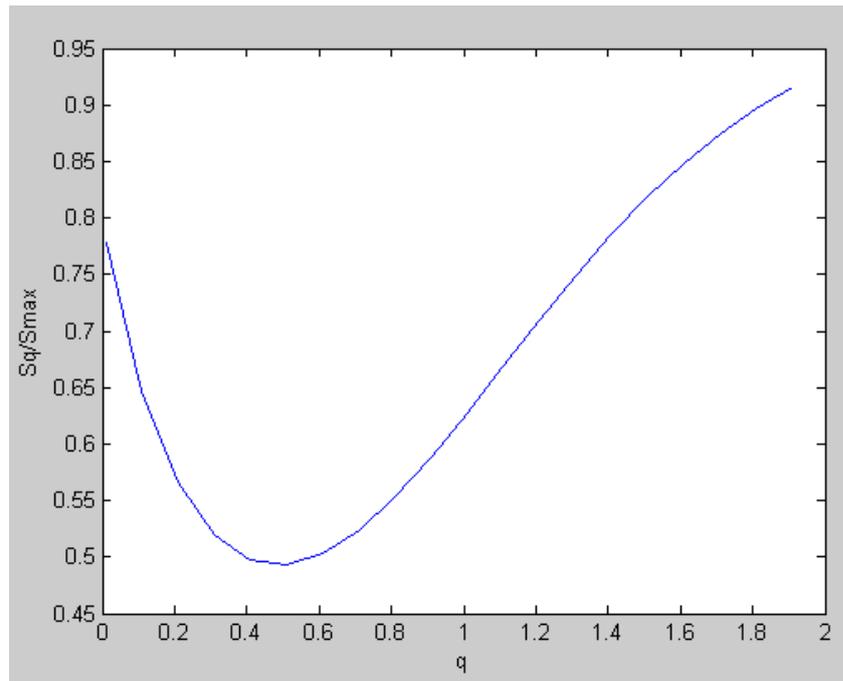


Figura 2.4: Adaptada de (Rodrigues e Giraldi, 2009b). Cálculo automático do valor de  $q$ .

$[0.01, 0.02, \dots, 2.0]$ , e verificando para qual valor esta relação é minimizada. Na Figura 2.4, pode observar-se como a relação entre  $S_q$  e  $S_{MAX}$  se comporta com a variação do valor de  $q$ .

Além da entropia não-extensiva, é possível o uso do algoritmo  $k$ -means para executar a segmentação. Este é um método de classificação muito conhecido. Apesar de ser publicado em 1967, (Macqueen, 1967), e de que, desde então já surgiram diversos outros algoritmos de clusteirização, esta técnica ainda é muito utilizada (Jain, 2009). Neste algoritmo, inicialmente são definidos  $k$  centros de *clusters*. Em seguida, todos os elementos são agrupados nos clusters cujos centróides estão mais próximos. Então, o centróide de cada *cluster* é recalculado e volta-se a reagrupar os elementos e recalculando o centróide até que o algoritmo chegue a uma convergência ou até que seja atingido um critério de parada.

O  $k$ -Means também pode ser utilizado como técnica de segmentação, separando uma imagem em  $k$  *clusters* segundo a Equação (2.24), encontrada no livro de Forsyth e Ponce (Forsyth e Ponce, 2002). Este algoritmo foi utilizado em muitos trabalhos de processamento de imagens agrupando elementos com características similares em classes de diversos gêneros (Niemisto et al., 2007), (Wang e Leaser, 2007), (Bastos et al., 2008), (Zhu et al., 2009), (Wu et al., 2009).

$$\Phi(\text{clusters}, \text{data}) = \sum_{i \in \text{clusters}}^k \left\{ \sum_{j \in \text{cluster } i} (x_j - c_i)^T (x_j - c_i) \right\} \quad (2.24)$$

onde:  $\Phi(\text{clusters}, \text{data})$  é a variância intra-cluster;  $c_i$  é o centro do *cluster*  $i$ ;  $x_j$  é o elemento  $j$  a ser clusterizado e  $k$  é a quantidade de clusters.

Outra técnica de segmentação que pode ser utilizada é a de modelos deformáveis ou *snakes*. Proposta por Michael Kass em 1988 em (Kass et al., 1988), uma *snake* é uma forma fechada modelada através da minimização de sua função de energia descrita na Equação (2.25).

$$E = E_{int} + E_{ext} \quad (2.25)$$

onde:  $E$  é Função de energia a ser minimizada;  $E_{int}$  é energia interna e  $E_{ext}$  é a energia externa.

A definição de modelos deformáveis como algoritmo segmentador é controversa, uma vez que sua execução já resulta no objeto de interesse, e não apenas em regiões com características similares como outros métodos de segmentação. Em 2004, os autores Camps e Sznaiier utilizaram *snakes* para segmentar partes de objetos, tornando seu algoritmo mais robusto em relação à oclusão (Gentile et al., 2004). Em 2005, Tissainayagam e Suter utilizaram modelos deformáveis para selecionar o objeto de interesse e seus pontos-chave em um método que funciona bem para objetos que se diferenciam bastante do *background* (Tissainayagama e Suterb, 2005). Ainda em 2005, Giraldi e Rodrigues fizeram um estudo para melhorar a inicialização, convergência e a utilização de memória do algoritmo de modelos deformáveis (Giraldi et al., 2005).

Cada uma das técnicas de segmentação citadas anteriormente oferece resultados diferentes para uma mesma aplicação. Na Figura 2.5, pode-se verificar a comparação dos resultados da segmentação através de limiar adaptativo (Figura 2.5b),  $k$ -means (Figura 2.5c), entropia clássica (Figura 2.5d) e modelos deformáveis (Figura 2.5e). A segmentação por limiar adaptativo ofereceu os melhores resultados para este tipo de objeto. As técnicas de entropia e  $k$ -means tiveram um resultado bom em apenas alguns casos. A segmentação através de modelos deformáveis apresentou bons resultados, porém seu desempenho muda conforme o ponto de inicialização selecionado. Observando a Figura 2.5, pode-se notar que existem múltiplas possibilidades de resultados, o que é um dos fatores de maior problema para sistemas que utilizam técnicas de segmentação como este trabalho.

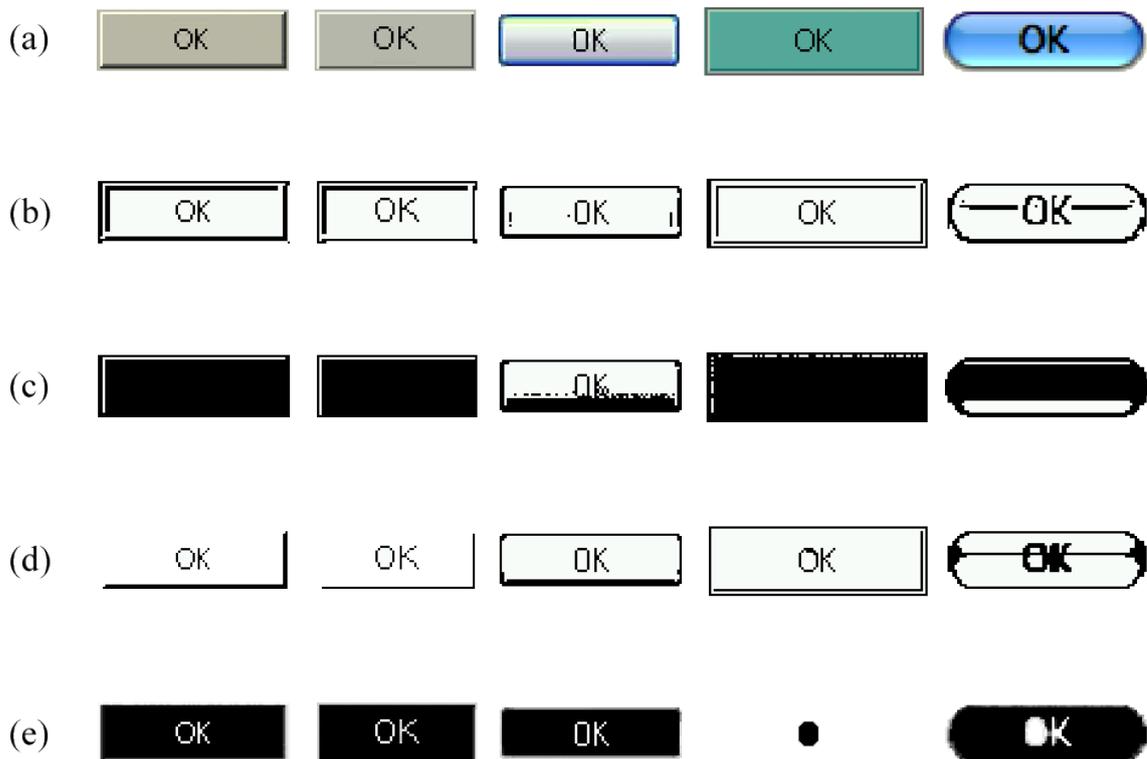


Figura 2.5: Comparação entre os algoritmos de segmentação estudados. Em (a), cinco imagens de botões. Em (b),(c),(d) e (e), o resultados das execuções dos algoritmos de segmentações das imagens utilizando respectivamente limiar adaptativo,  $k$ -means, entropia clássica e modelos deformáveis.

### 2.3.2 Detecção de Bordas

A segmentação de imagens tem como objetivo agrupar pixels com características similares em regiões. A detecção de bordas por sua vez procura as fronteiras entre as regiões. Bordas de objetos tendem a ser descontinuidades de intensidades em imagens (Ballard e Brown, 1982). Estas descontinuidades ou variações de intensidades podem ser de diversos tipos diferentes conforme pode-se observar na Figura 2.6. Em imagens naturais, pode-se encontrar diversos tipos de bordas como gradiente, bilinear ou ruidoso, como nos exemplos nas Figuras 2.6b, 2.6c e 2.6d. Em imagens sintéticas, por não terem ruído, as bordas dos objetos tendem a ser compostas, como na Figura 2.6d.

Na maioria das imagens, as diferenças entre as intensidades dos pixels aumenta gradativamente como na Figura 2.7 (Gonzalez e Woods, 2001). Quanto mais longo for o gradiente, mais espessa é a borda. A segunda derivada da imagem gera dois picos para cada borda, um positivo e um negativo. O espaço entre estes picos define a borda.

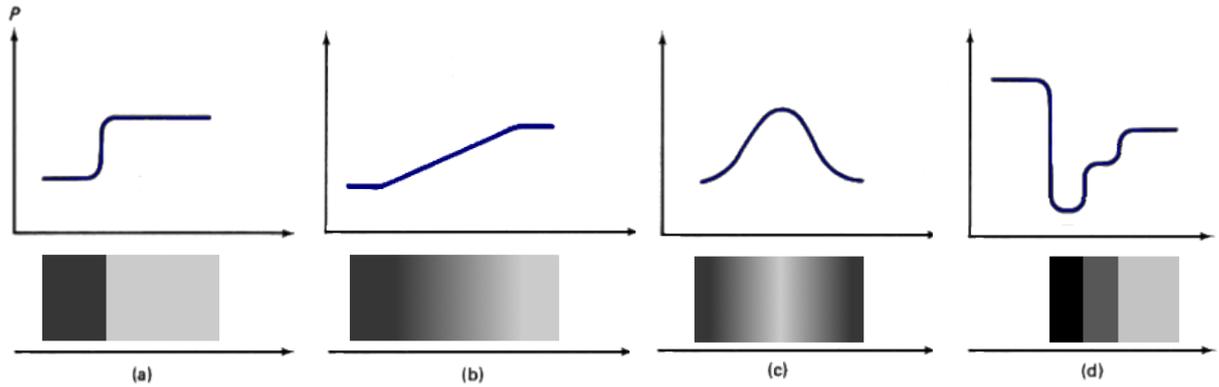


Figura 2.6: Adaptada de (Ballard e Brown, 1982). Tipos de variação de intensidade de pixels. Em (a) uma borda abrupta, em (b) um gradiente, em (c) um gradiente bilinear e em (d) uma borda composta.

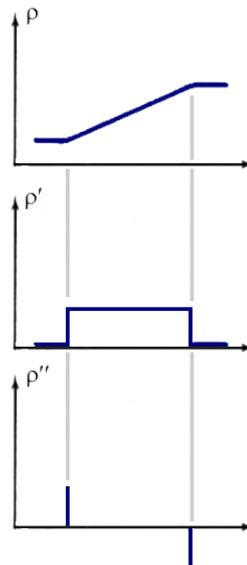


Figura 2.7: Primeira e segunda derivadas em uma linha transversal à uma borda.

Derivadas de primeira ordem em uma imagem são calculadas através de gradientes, enquanto que derivadas de segunda ordem são calculadas com o Laplaciano. Para computar uma derivada parcial de primeira ordem, pode-se utilizar os operadores de Canny, Roberts, Prewitt e Sobel, alguns deles demonstrados na Figura 2.8 (Forsyth e Ponce, 2002).

O mapa de bordas de uma imagem é obtido através da cálculo da convolução entre o operador ou *kernel* desejado e a imagem a ser filtrada. Em cada par de *kernels* das Figuras 2.8a e 2.8b, o filtro à esquerda detecta as bordas horizontais enquanto que o filtro à direita responde às bordas verticais. Na Figura 2.8c, os filtros respondem às bordas de  $-45^\circ$  e  $45^\circ$  respectivamente.



A Figura 2.9 mostra que existe pouca diferença entre os operadores de borda comparados. É importante ressaltar porém, que os filtros de *Prewitt* e *Sobel* capturam mais detalhes enquanto que o filtro de *Canny* detecta as bordas mais significativas das imagens se mostrando assim mais robusto em relação a ruído.

### 2.3.3 Extração de Atributos

Até aqui foram vistas as técnicas para separação dos objetos de interesse em uma imagem digital. Esta seção se dedica à extração das propriedades, também conhecidas por *features* ou atributos destes objetos. Através destas características, estes objetos podem ser classificados em diferentes classes por meio de um processo conhecido como classificação. Existem diversas técnicas de seleção de atributos como (Ahmad e Dey, 2004) ou também (S. Gunal, 2008), porém a seleção dos atributos a serem extraídos está diretamente ligada ao objetivo a ser alcançado e a melhor forma para seleção dessas características ainda é por meio de execução de diversos experimentos.

Dentro de uma imagem, um objeto possui diversos atributos, sendo que alguns destes atributos podem ser mais expressivas do que outras de acordo para a classificação destes objetos. Essas características podem ser separadas de acordo com o tipo de informação que elas representam, de modo que podem existir atributos de cor, textura e geometria. Atributos de cor são todos aqueles cujas informações são inerente às cores do objeto, incluindo tonalidades de cinza. As propriedades de textura por sua vez, referem-se aos padrões de variações de cores e formas no objeto de interesse. Finalmente, os atributos de geometria representam dados como posição, forma e tamanho do objeto.

No problema estudado neste trabalho, algumas das características dos objetos podem mudar quando ocorre mudança de cena, dificultando assim o seu reconhecimento. Alguns atributos podem ter alterações mais ou menos significativas, de acordo com o tipo de objeto em cena. Em uma imagem natural, por exemplo, uma árvore pode ter sua forma alterada em duas imagens diferentes se houver uma grande mudança no ângulo da foto, mas as cores e a textura predominantes sofrem menor alteração, o que torna estes atributos mais significativos para este caso.

A classificação de um objeto como pertencente a uma determinada classe, restringe quais características serão comparadas entre os objetos, diminuindo assim a quantidade de distâncias que

devem ser calculadas e tornando o algoritmo de *matching* mais rápido. Além disto, o conhecimento da classe na qual o objeto de interesse pertence fornece algumas informações *a priori* sobre o comportamento deste objeto e das variações que ele pode sofrer.

## 2.4 Classificação

A classificação de objetos pode ser feita de forma supervisionada e não-supervisionada. Neste trabalho será adotada a primeira maneira, que consiste em organizar objetos em grupos previamente definidos por meio de treinamento ou configuração do classificador. Este agrupamento ocorre de acordo com a similaridade entre as características dos objetos e dos grupos. Estes atributos podem ser interpretados como uma forma de representação destes objetos, uma vez que eles contemplam as informações mais pertinentes dos mesmos.

O algoritmo de classificação é treinado ou configurado de forma supervisionada com um conjunto de imagens de controle e as classes a que estas imagens pertencem. Após o treinamento, outras imagens, diferentes das imagens de controle, porém pertencentes às mesmas classes, são separadas em grupos pelo algoritmo. A porcentagem de imagens classificadas de forma correta é definida como resultado ou desempenho do classificador.

Existem diversos algoritmos para classificar imagens, cada um deles possuindo vantagens e desvantagens em relação ao tipo de dado a ser classificado, tempo de execução, quantidade de dimensões, quantidade de classes, etc. Em (Walt e Barnard, 2007), os autores estudaram a relação entre o resultado de cada algoritmo de classificação e os tipos de dados a serem classificados. Alguns dos algoritmos de classificação mais comuns são as redes Bayesianas (Jensen e Nielsen, 2007), redes neurais (Bishop, 1995) e *k*-means (Macqueen, 1967).

### 2.4.1 Classificação com Redes Bayesianas

A classificação de elementos pode ser feita utilizando-se Redes Bayesianas, que são redes causais onde o peso da conexão causal é representado como uma probabilidade condicional (Jensen e Nielsen, 2007), Também são conhecidas como redes de crença e podem ser descritas como diagramas de eventos relacionados entre si onde esta relação é expressa conforme a probabilidade de

um evento causar o outro. Ou seja, quando um evento tem diversas causas possíveis, estas causas estão ligadas a este evento de acordo com a probabilidade de estas causas ocorrerem.

Após a configuração da rede, é possível inferir a probabilidade de um fato ocorrer a partir do conhecimento de outros fatos. A Figura 2.10 mostra uma rede Bayesiana de atributos e classes. Como classificador, os atributos de objetos podem ser representadas como causas e as classes como efeito.

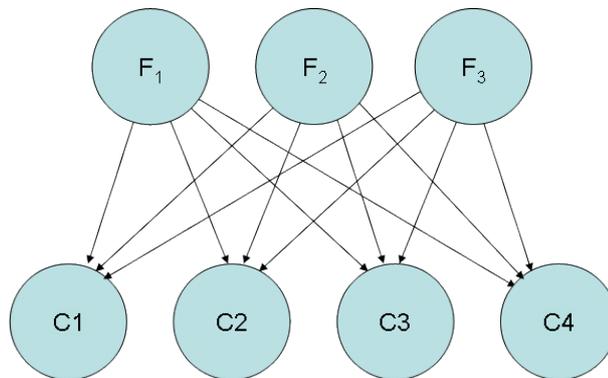


Figura 2.10: Rede Bayesiana de um classificador.  $F_1$ ,  $F_2$  e  $F_3$  são atributos e  $C_1$ ,  $C_2$ ,  $C_3$  e  $C_4$  são classes.

A classificação de um objeto em uma classe é obtida através da comparação das probabilidades das classes  $C_1$ ,  $C_2$ ,  $C_3$  ou  $C_4$  ocorrerem dado que ocorreram os atributos  $F_1$ ,  $F_2$ ,  $F_3$  com as intensidades  $f_1$ ,  $f_2$  e  $f_3$  respectivamente. A classe do objeto será aquela que possuir a maior probabilidade  $P$  de ocorrer. Utiliza-se a seguinte notação mostrada na Equação (2.26) para representar o cálculo de  $P$  para as classes  $C_1$ ,  $C_2$ ,  $C_3$  e  $C_4$ .

$$P(C_x | F_1 = f_1, F_2 = f_2, F_3 = f_3) \quad (2.26)$$

$$C_x = \{C_1, C_2, C_3, C_4\} \quad (2.27)$$

Para poder calcular as probabilidades das classes, são necessárias as probabilidades condicionais entre os atributos e as classes. O levantamento destas probabilidades ou configuração da Rede Bayesiana é feito a partir de um conjunto de objetos previamente separados em classes. Este levantamento é realizado através da contagem das probabilidades de cada atributo de cada item para cada classe.

## 2.4.2 Classificação com Redes Neurais

Ao contrário das redes Bayesianas, as redes neurais não necessitam de um levantamento estatístico de cada atributo em cada classe. Neste método, o treinamento da rede é realizado através da entrada de dados de controle e seus resultados esperados.

Uma rede neural é composta por vários neurônios artificiais que possuem diversas entradas, um peso para cada uma das entradas, uma função de ativação e uma saída. Estes neurônios podem ser agrupados em diferentes topologias, com nenhuma, uma ou diversas camadas intermediárias de acordo com o tipo de problema a ser resolvido. Para cada conjunto de entradas e resultados, os pesos das entradas destes neurônios são reajustados de acordo com o erro entre o resultado produzido e o resultado esperado.

O reajuste dos pesos de uma rede neural pode ser feito de diversas formas diferentes de acordo com a topologia da rede. No método *backpropagation*, descrito por (Bishop, 1995), o reajuste do peso  $w_{ij}$  do neurônio  $i$  na camada  $j$  em um neurônio de entrada  $x_i$  é dado pela Equação (2.28):

$$\Delta w_{ji} = -\eta \delta_j x_i \quad (2.28)$$

onde:  $\eta$  é a taxa de aprendizado;  $\delta_j$  é o erro na camada  $j$  e  $x_i$  é a entrada  $i$ .

## 2.4.3 Similaridade Entre Objetos

Além das redes neurais e Bayesianas, a classificação de imagens ainda pode ser feita utilizando-se *k*-means, método descrito anteriormente na Seção 2.3.1. O *k*-means é um algoritmo de classificação não supervisionado, de modo que não requer treinamento a partir de uma base de dados inicial. Por este motivo, o resultado de sua classificação nem sempre é o esperado. Uma alternativa para este problema é a utilização de conhecimento prévio das classes para proporcionar melhores resultados na classificação de objetos. Pode-se por exemplo extrair alguns atributos das imagens e os utilizar para separar os *clusters* de acordo com sua textura. (Bastos et al., 2008)

Assim, a classificação pode ser realizada por meio da comparação entre as propriedades do objeto a ser classificado e as médias das propriedades dos objetos na base de aprendizagem. O grau de similaridade entre essas características define a classe do objeto. Porém, a classificação através da comparação de somente um atributo não é suficiente, deve-se considerar vários os atributos do objeto, cada um com seu peso.

Os atributos extraídos de cada objeto constituem um vetor que representa de forma resumida o objeto do qual extraído. Este processo é conhecido como quantização vetorial. A classificação é dada pela menor medida de distância entre os vetores. Quanto mais similares são os vetores, menor é a distância entre eles. Um valor alto para a distância pode significar grandes diferenças entre os objetos.

Existem diversas técnicas para a o cálculo de similaridade entre dois vetores. As principais classes de medidas são as de cor, textura e forma (Shapiro e Stockman, 2000). Porém, com a representação de todos estes atributos em um vetor, é possível fazer uso de qualquer medida de distância para cálculo de similaridade entre vetores. A medida de similaridade entre vetores pode ser calculada através da distância de *Minkowski* e da distância de *Kullback-Leiber*, também conhecida por entropia relativa (Rodrigues, 2003).

A distância de *Minkowski* é uma generalização da distância euclidiana e é sensível à deslocamentos de histograma, fenômeno este, que ocorre principalmente quando há alteração de iluminação ou ruído (Rodrigues, 2003). Em imagens sintéticas, como as estudadas neste trabalho, estes dois fatos ocorrem em menor escala, viabilizando o uso desta técnica. A distância de *Minkowski* entre dois vetores  $p$  e  $q$  é calculada através da Equação (2.29).

$$D_M(p, q) = \left[ \sum_{i=1}^n |p_i - q_i|^x \right]^{1/x} \quad (2.29)$$

O cálculo da similaridade entre dois vetores através das distâncias euclidiana e de *Minkowski* possui uma fragilidade conhecida como erro da normalização relativa que ocorre quando o espaço vetorial envolve vários atributos heterogêneos que, embora normalizados, os atributos com menores intensidades podem ser suplantados por aqueles nas maiores faixas, independentemente de sua capacidade discriminativa.

Formalizando, pode-se calcular a distância dois vetores  $\partial = [f_1, f_2, \dots, f_n]$  e  $\partial' = [f'_1, f'_2, \dots, f'_n]$ , onde  $f_i, f'_i \in [0, 1]$ . Denomina-se a faixa de atributos com menores valores por  $\epsilon_A$  e a faixa com maiores valores por  $\epsilon_B$ . Se  $\epsilon_A$  for suficientemente próximo de 0 e  $\epsilon_B$  suficientemente próximo de 1, a distância euclidiana, será principalmente induzida por  $\epsilon_B$ . Se, por acaso, os vetores associados ao espaço  $\epsilon_A$  forem os mais discriminantes, isso torna os vetores associados a  $\epsilon_B$  falsos discriminantes.

Uma possível solução para este problema é composta da utilização de atributos representados por distribuições probabilísticas ao invés de valores absolutos e do cálculo da distância entre duas distribuições probabilísticas  $f$  e  $f'$  através da entropia relativa ou divergência de *Kullback-Leiber* conforme a Equação (2.30) (Kullback e Leiber, 1951):

$$Div_{KL}(f||f') = \sum_x f(x) \log \frac{f(x)}{f'(x)} \quad (2.30)$$

Aplicando-se a generalização de Tsallis à Equação (2.30) chega-se a forma da divergência não-extensiva de Kullback-Leiber como definida na Equação (3.2) (Tavares, 2003):

$$Div_{KLq}(f||f') = \sum_x f(x) \log_q \frac{f(x)}{f'(x)} \quad (2.31)$$

É possível notar que  $Div_{KL}(f||f')$  é diferente de  $Div_{KL}(f'||f)$ , e assim não pode ser considerada uma distancia. Assim, faz-se uso da soma destas divergências para definição de uma métrica que seja igual, independentemente da ordem de seus fatores, dada pela Equação (2.32). A essa métrica damos o nome de Distância de *Kullback-Leiber* (Tavares, 2003).

$$Dist_{KL}(f||f') = \frac{1}{2} (Div_{KL}(f||f') + Div_{KL}(f'||f)) \quad (2.32)$$

A Equação (2.30) trata da divergência entre duas distribuições probabilísticas. Quando existem mais de uma distribuição probabilística com interdependência entre elas, o cálculo da divergência entre dois vetores é realizado através do uso da regra da cadeia aplicada à Equação (2.30) e possui complexidade  $O(n^{|f|})$  para uma quantidade de atributos  $n$  com tamanho  $|f|$ . O alto custo computacional inviabiliza o uso deste algoritmo em tempo real. Porém, neste trabalho vamos assumir de que não há interdependência entre os atributos de forma que é possível a utilização da somatória das distâncias entre as distribuições probabilísticas como medida de similaridade entre objetos.

Sendo assim, a distância entre dois vetores de distribuições probabilísticas  $p = \{f_1, f_2, \dots, f_n\}$  e  $q = \{f'_1, f'_2, \dots, f'_n\}$ , com um total de  $n$  distribuições cada, é obtida através da soma das distâncias entre cada uma de suas distribuições conforme Equação (2.33).

$$D(p, q) = \sum_{x=1}^n D_{KL}(f_x || f'_x) \quad (2.33)$$

Este algoritmo tem complexidade  $O(n|f|)$  para uma quantidade de atributos  $n$  com tamanho  $|f|$  e seu resultado tem domínio em  $\{0, n\}$ , uma vez que cada  $D_{KL}(f_x || f'_x)$  pode assumir valores entre zero e um.

O resultado da classificação de um objeto é a definição do conjunto de elementos ao qual o objeto pertence. Por exemplo: ao final desta etapa, um objeto pode ser classificado como um ícone, botão, caixa ou área de texto. A classificação permite limitar os atributos que serão comparados no momento do reconhecimento, e fornece um conhecimento das variações que os objetos podem sofrer.

## 2.5 Conclusões

Neste capítulo foram descritas algumas das técnicas mais comuns em processamento de imagens digitais. A solução do problema proposto neste trabalho envolve o uso de algumas destas técnicas, desde a visão de baixo nível como detecção de bordas passando pela visão de médio nível como a segmentação de imagens, bem como a visão de alto nível como classificação e reconhecimento de objetos. O próximo capítulo discute como as técnicas estudadas aqui podem ser utilizadas para a detecção dos objetos em interfaces gráficas, a extração de suas propriedades e sua classificação.

## Capítulo 3

# PROPOSTA DE TRABALHO

Neste capítulo é descrita a metodologia utilizada para a realização de duas tarefas: a detecção e a classificação de objetos sintéticos. Os algoritmos capazes de executar estas tarefas serão a base de nosso automatizador de tarefas hipotético que é capaz de gravar e reproduzir uma sequência de interações com os objetos do sistema operacional de forma robusta à variação de posição e a pequenas variações de forma, cor e textura.

### 3.1 Objetos de Detecção

Em uma interface gráfica, existem diversos tipos diferentes de objetos. Cada um deles possui características singulares que os diferenciam dos demais. A Tabela 3.1 relaciona cada um dos objetos estudados e seus principais atributos.

Tabela 3.1: Objetos em Interfaces Gráficas

<b>Objetos</b>	<b>Interações</b>	<b>Características</b>
Botões	Cliques	Cor de fundo predominante Bordas Uniformes Aspecto retangular
Ícones	Cliques	Grande variação de cor Grande variação de forma Bordas ruidosas
Caixas de Texto	Cliques e Digitação	Possuem somente texto Pouca variação de altura Fundo predominantemente branco
Áreas de Texto	Cliques e Digitação	Podem possuir barras de rolagem Podem possuir texto Podem possuir várias linhas

### 3.1.1 Botões

Botões (Figura 3.1a) geralmente são retangulares, possuem uma única cor e textos simples no centro. Porém, estas características não são predominantes e podem mudar na medida em que novas versões do programa-alvo forem desenvolvidas. A forma pode mudar de um retângulo com ângulos retos para outro com bordas suavizadas. A cor de fundo também pode mudar, deixando de ser homogênea para ser um *degradê* de cores. O texto do botão também pode variar de diversas formas, desde a aplicação de *anti-aliasing*, alteração do tipo de fonte ou substituição do texto por uma palavra mais adequada a sua função.

### 3.1.2 Ícones

Ícones (Figura 3.1b) são pequenas figuras com um conteúdo semântico bem definido, cuja imagem está diretamente relacionada com a ação executada pelo sistema quando o objeto é ativado. Geralmente, esses objetos não variam muito em forma, cor ou textura, mas quando são trocados (em uma nova versão do programa-alvo, por exemplo), podem ter todas as suas propriedades completamente alteradas de modo a inviabilizar seu reconhecimento, a menos que seja analisado seu significado semântico. Porém, neste trabalho, não será estudada a interpretação do significado semântico de cada ícone uma vez que este tema, por si só, é suficientemente complexo para ser estudado de forma exclusiva.

### 3.1.3 Caixas de Texto

Caixas de Texto (Figura 3.1c), assim como os botões, são em sua maioria objetos retangulares e sua cor de fundo mais comum é branco e, geralmente, não possuem texto antes de receberem dados de entrada. As alterações que estes objetos podem sofrer são variações em sua largura, variação do tipo de fonte do texto e o uso de *anti-aliasing* para as fontes. O maior obstáculo para a detecção correta deste objeto porém é a contextualização, ou seja, a capacidade do algoritmo de associar uma caixa de texto aos objetos vizinhos a ela quando existe mais de um destes elementos na tela. Este fato é comum em formulários onde se possui várias caixas de texto com características visuais exatamente iguais, porém, os dados que cada uma representa são definidos pelos textos ao lado das caixas.

### 3.1.4 Áreas de Texto

Áreas de Texto (Figura 3.1d) são objetos dentro dos quais podem existir textos longos. Este tipo de objeto geralmente possui fundo branco, assim como as caixas de texto. Dentre as alterações que as áreas de texto podem sofrer estão as variações de largura e altura, conforme a janela que envolve a área de texto em questão é redimensionada, maximizada ou restaurada. Além disto, uma área de texto pode ter seu conteúdo interno em fontes de tipos, tamanhos e cores diferentes.

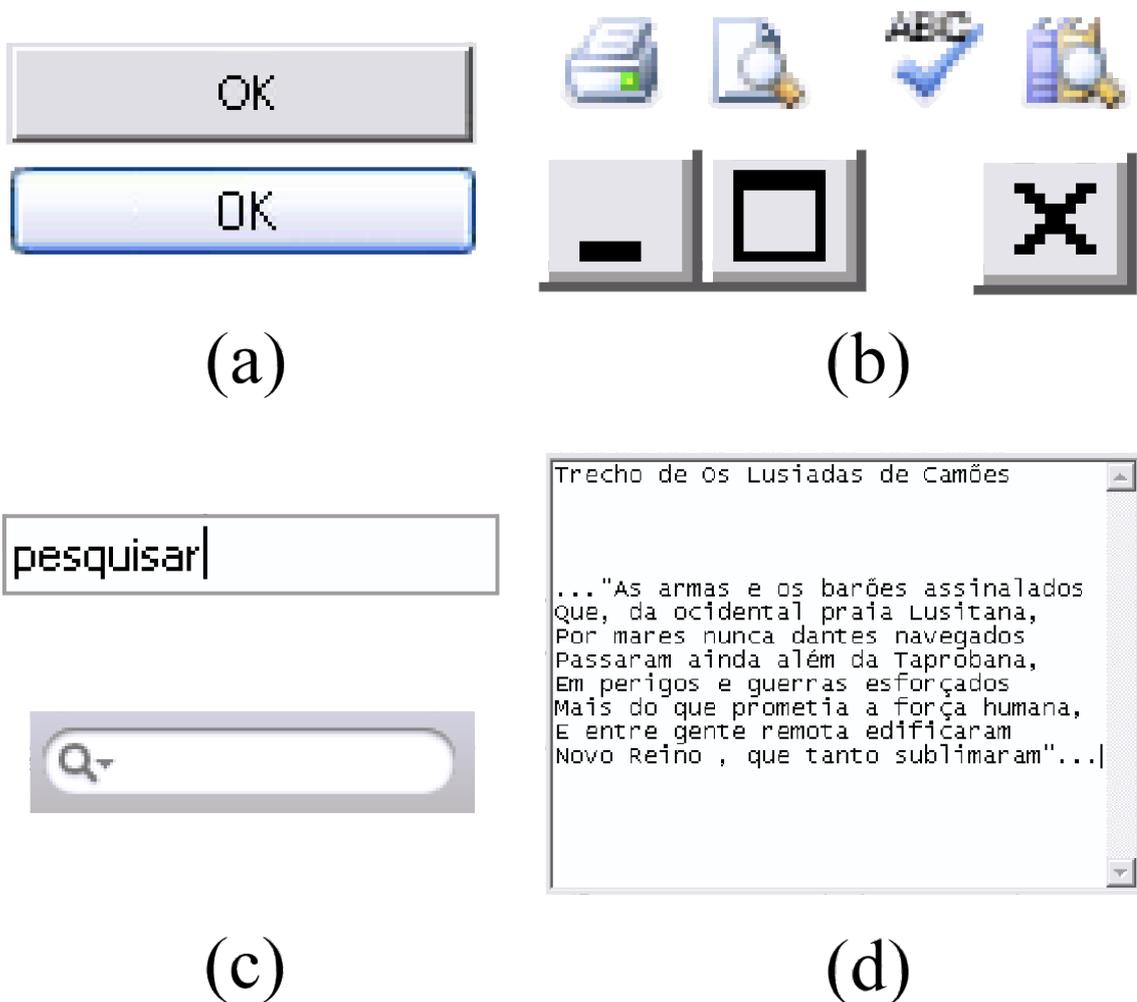


Figura 3.1: Exemplos de objetos em interfaces gráficas. botões em (a), ícones em (b), caixas de texto em (c), uma área de Texto (d) e um *container* com textos, botões e ícones (e).

## 3.2 Visão Geral do Método Proposto

Para automatizar uma sequência de passos em um programa são necessárias duas etapas: a etapa de aprendizado e a de reprodução. Na primeira fase, também conhecida como filmagem ou gravação, todas as interações com os objetos são salvas bem como as características destes objetos. Na etapa seguinte, conhecida como reprodução ou execução, a ferramenta de automação deve repetir as interações com os mesmos objetos utilizados durante a filmagem.

Encontrar objetos na etapa de reprodução, é a principal dificuldade das ferramentas de automação baseadas em componentes, uma vez que algumas características do sistema operacional podem ser alteradas entre a gravação e a reprodução. Essas alterações podem ocorrer devido a instalação de programas ou atualizações que alteram bibliotecas utilizadas pelo sistema. Além disto, os objetos podem sofrer diversas alterações como mudança de posição, de forma, de escala e até oclusão entre a gravação e o momento da reprodução. Um ícone, por exemplo, somente pode sofrer translação enquanto que uma caixa de texto pode ter variação de posição ou de largura. Uma área de texto, por sua vez, pode variar de largura, altura ou posição.

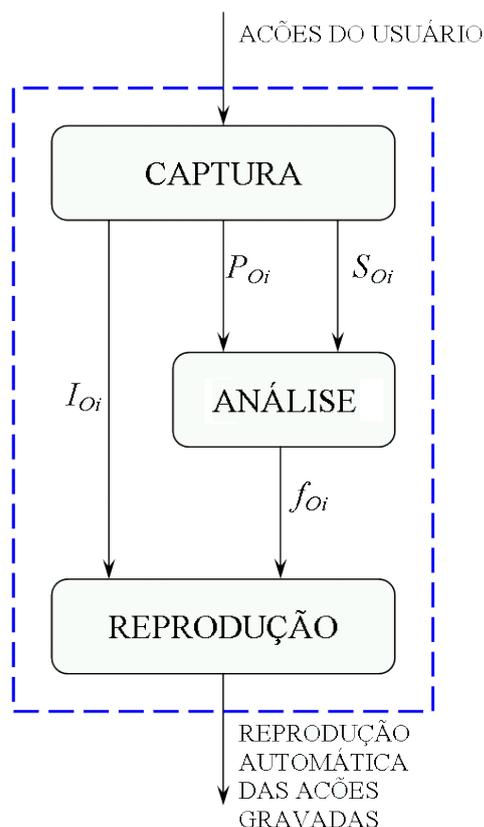


Figura 3.2: Etapas do programa de automação proposto e os vetores resultantes ao final de cada etapa. O processo inteiro deve ser capaz de capturar as interações de um usuário e reproduzi-las de forma robusta ao final.

A proposta de detecção dos objetos por meio de suas propriedades visuais procura minimizar estes efeitos. Para isto, além da gravação e execução, existe ainda uma fase adicional, onde os dados coletados são analisados. As três etapas estão demonstradas na Figura 3.2.

Na primeira etapa, denominada Captura, para cada interação  $i$  realizada pelo usuário, são armazenadas as coordenadas do mouse no instante do clique e a digitação do teclado em um vetor  $I_i$  enquanto a captura de tela no momento exato da interação é gravado em  $S_i$ , assim como as coordenadas do mouse preenchem o vetor  $P_i$ .

A segunda etapa deste processo é conhecida como Análise. Nesta etapa, cada captura de tela é analisada em conjunto com a coordenada do mouse naquele instante. Essa análise permite o isolamento do objeto  $O_i$  com o qual houve a interação  $i$ . Com o objeto  $O_i$  separado do resto da imagem, é criado um vetor  $f_{O_i} = \{c_1, c_2, \dots, c_3\}$  com as características que foram extraídas do objeto. A partir deste momento,  $f_{O_i}$  irá representar, de forma quantizada, o objeto  $O_i$ . Com  $f_{O_i}$  é possível fazer a classificação de  $O_i$  em uma das classes descritas na Tabela 3.2.

Tabela 3.2: Classes de Objetos

<b>Classe</b>	<b>Objetos</b>
C1	Botões
C2	Ícones
C3	Caixas de Texto
C4	Áreas de Texto

Na terceira etapa, chamada de Reprodução, para cada uma das  $i$  interações que foram gravadas,  $O_i$  é procurado na tela por meio da comparação de  $f_{O_i}$  com as características extraídas de todos os objetos que são exibidos na tela. Quando  $O_i$  é encontrado, sofre a mesma interação  $I_i$  definida durante a gravação.

## 3.3 Técnicas Utilizadas

### 3.3.1 Separação do Objeto de Interesse

A separação do objeto de interesse  $O_i$  com o qual houve interação é a parte mais crítica desta proposta. Uma boa separação inicial proporciona informações mais apuradas sobre os objetos e, conseqüentemente, torna o *matching* mais preciso.

Durante a etapa de gravação, são criados três vetores: o vetor de *screenshots*  $S_i = \{s_1, s_2, \dots, s_i\}$ , o vetor de coordenadas  $P_i = \{p_1, p_2, \dots, p_i\}$  e o vetor de ações realizadas  $I_i = \{a_1, a_2, \dots, a_i\}$  onde são guardadas as informações de cada ação realizada pelo usuário. Para cada interação  $i$  gravada, são gravados um *screenshot*  $S$ , uma coordenada  $P$  e uma ação  $I$ .

A primeira tarefa a ser realizada na etapa de análise é a extração do objeto  $O_i$ . Inicialmente, tem-se a imagem  $S_i$  e a coordenada de interação  $P_i$  como pode-se observar na Figura 3.3. Em seguida, a imagem  $S_i$  é segmentada (Figura 3.4) e então aplica-se um algoritmo de detecção de contornos que resulta em um vetor com as coordenadas de cada contorno encontrado  $V_c$  que envolve  $P_i$ . Estes contornos são filtrados por uma função de seleção condicional que verifica as coordenadas de cada contorno que engloba  $P_i$  (Figura 3.5). Os resultados desta etapa são as coordenadas dos limites de  $O_i$  como pode-se observar na Figura 3.6.

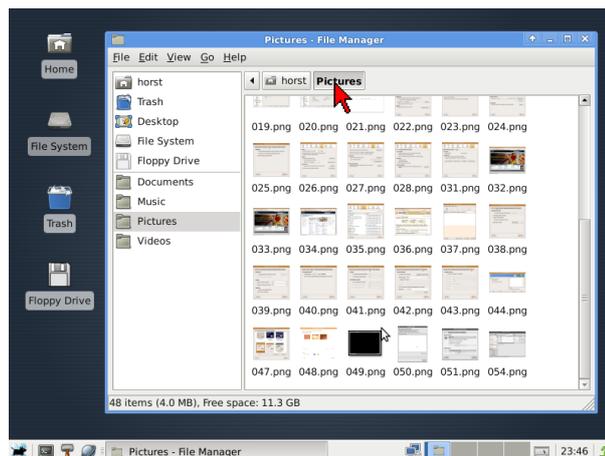


Figura 3.3: Imagem  $S_i$ . A coordenada de interação  $P_i$  está representada pelo *cursor* vermelho

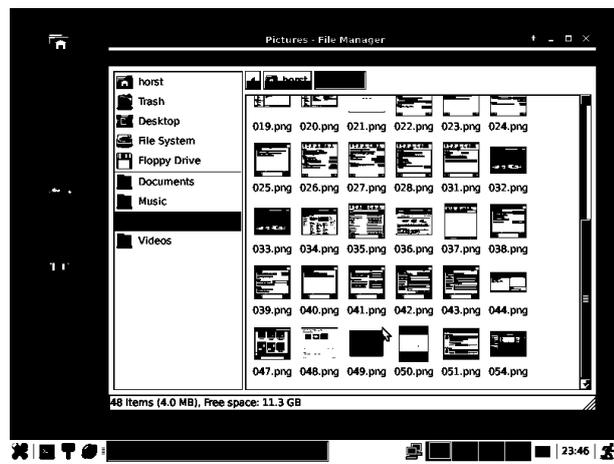


Figura 3.4: Imagem da Figura 3.3 segmentada com entropia de Tsallis Estendida com valores  $q_1 = 1,5$  e  $\alpha = 1,1$



Figura 3.5: Contorno ao redor da coordenada de interação filtrado.

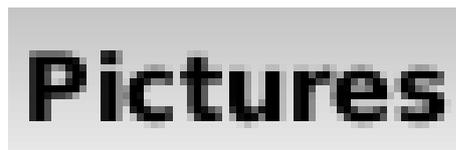


Figura 3.6: Objeto extraído ao final da primeira etapa

### 3.3.2 Base de Dados

Para análise dos algoritmos de reconhecimento e classificação de imagens foram utilizadas duas bases de dados com imagem de controle, também conhecidas como *Ground Truth*. As imagens dessas bases serão utilizadas para definição dos parâmetros de execução dos algoritmos e para validação de seus desempenhos.

Para criação da base de dados que será utilizada neste trabalho, foram extraídas capturas de telas de três fontes de dados independentes, cada uma com um sistema operacional. Foi criada uma rotina de navegação aleatória que abre cada aplicativo e simula o clique do mouse em posições aleatórias para extrair o *screenshot* quando ocorre uma alteração na tela. A Tabela 3.3 descreve os sistemas dos quais as capturas de tela foram extraídas e a quantidade de capturas de tela retiradas cada sistema.

Tabela 3.3: Fontes de dados

Fonte	Sistema Operacional	Qtd. Capturas de Tela
1	Ubuntu Linux	200
2	Microsoft Windows 2000	250
3	Microsoft Windows XP	350

Foram criadas duas bases de imagens, a primeira delas para avaliação dos desempenhos dos diferentes algoritmos de segmentação, chamada de Base A ou Base de Capturas. Esta base possui duas mil imagens de capturas de tela onde cada imagem  $P$  está associada a uma coordenada  $c$  e a uma outra imagem  $R$  que representa o objeto que deve ser detectado se a algoritmo estiver correto. A Figura 3.7 mostra a organização desta base de imagens. A coluna (a) da Figura 3.7 demonstra alguns exemplos de capturas de tela  $P$ . A coluna (b) da Figura 3.7 demonstra a coordenada de interação  $c$  que é armazenada em um arquivo de dados, associando-a com a imagem e com o objeto esperado  $R$ .

Para a classificação dos objetos de interesse é necessária uma base de dados com exemplos de objetos de cada *cluster*. O desempenho de um algoritmo classificador supervisionado depende diretamente da base de dados utilizada para seu treinamento ou parametrização. Neste aspecto, o ideal é uma base de dados com grande volume de objetos e bastante variação entre objetos da mesma classe de modo a representar a maioria dos objetos encontrados em situações reais. Assim sendo, além da Base de Capturas, foi criada também, uma base de imagens de objetos já extraídos.

Chamada de Base B ou Base de Objetos, este coletânea de imagens tem como propósito a extração de atributos comuns aos objetos de cada classe. Esta base de imagens possui 4 classes de objetos: Botões, Ícones, Caixas de Texto e Áreas de Texto, como pode-se observar na Figura 3.8.



Figura 3.7: Base de Capturas. Além das capturas de tela (a), a base de imagens conta também com uma coordenada de interação (b) e a imagem do objeto esperado após a detecção (c).

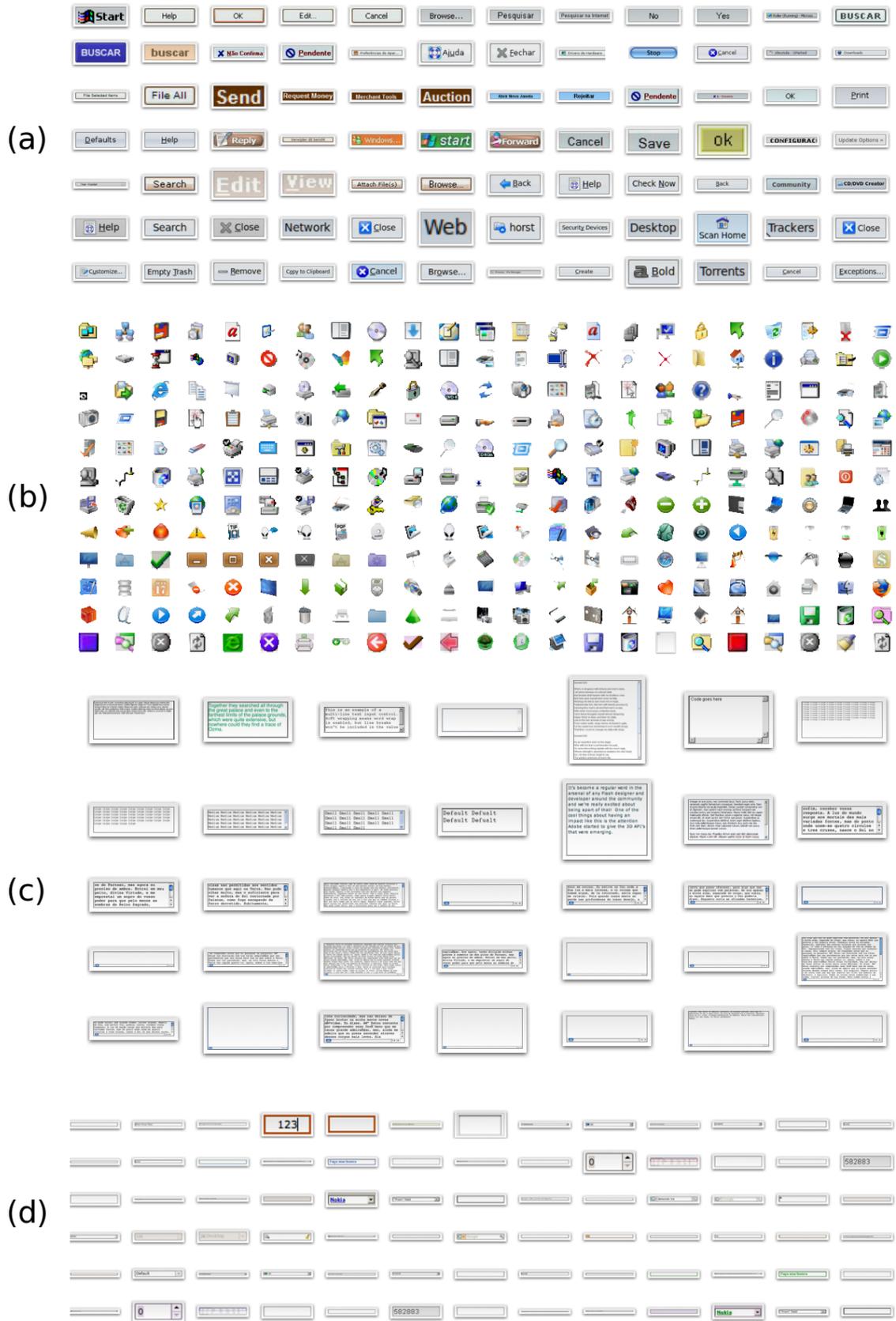


Figura 3.8: Base de Objetos. Botões em (a), Ícones em (b), Áreas de Texto em (c) e Caixas de Texto em (d).

### 3.3.3 Características Extraídas dos Objetos

Após a extração de  $O_i$ , é feita a análise de suas características e criação de um vetor de representação  $f_{O_i}$  que conterá os valores de cada um dos atributos extraídos de  $O_i$ . A Tabela 3.4 descreve os atributos que serão extraídas para esta dissertação. Foram selecionadas características de cores, texturas, forma e relacionais de modo a representar  $O_i$  sob vários aspectos.

Tabela 3.4: Atributos extraídas dos objetos

Índice	Atributo	Tipo
F1	Histograma HSV 18-3-3	Cor
F2	Histograma da projeção vertical	Textura
F3	Histograma de raios	Forma
F4	Histograma do corte vertical	Textura
F5	Histograma do corte horizontal	Textura
F6	Histograma do corte diagonal	Textura

### 3.3.4 Classificação dos Objetos

Após a criação de  $f_{O_i}$  com as principais características de  $O_i$ , é possível utilizar uma função de clusteirização  $FC(O_i)$  para classificar  $O_i$  em uma das quatro classes listadas na Tabela 3.2 de acordo com os valores de  $f_{O_i}$ . A classificação será realizada por meio do cálculo da similaridade entre  $f_{O_i}$  e o centróide  $G_x$  de cada *cluster*.

A Figura 3.9 ilustra um exemplo de uma representação em três dimensões de objetos de clusters diferentes onde a classificação de um elemento novo é obtida por meio da menor distância entre seus atributos e os centros de cada classe. Neste trabalho serão considerados seis atributos para caracterizar um objeto e a similaridade entre  $f_{O_i}$  e cada  $G_x$  é calculada por meio da somatória da similaridade entre de cada atributo de  $f_{O_i}$  e  $G_x$ .

A similaridade entre dois vetores pode ser calculada por meio da comparação entre suas características  $f_{O_1}$  e  $f_{O_2}$ . Conforme discutido na Seção 2.4.3, a comparação entres dois vetores de distribuições probabilísticas  $f_{O_i} = \{f_{O_i}^1, f_{O_i}^2, \dots, f_{O_i}^n\}$  e  $f_{O^T} = \{f_{O^T}^1, f_{O^T}^2, \dots, f_{O^T}^n\}$  pode ser feita com o cálculo da distância de *Kullback-Leiber* conforme a Equação (3.1).

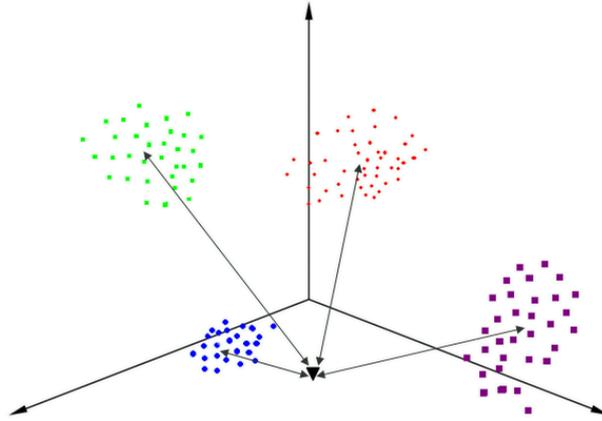


Figura 3.9: Exemplo de espaço tridimensional com a classificação de um objeto por meio do cálculo da simetria entre este objeto e os centros das classes.

$$D(f_{O_i}, f_{O^T}) = \sum_{x=1}^n D_{KL}(f_{O_i}^x \| f_{O^T}^x) \quad (3.1)$$

Na Seção 2.3.1, foi visto que Rodrigues e Giraldi propuseram uma extensão da entropia de Tsallis, utilizando dois parâmetros entrópicos,  $q_A$  e  $q_B$ , proporcionais às áreas de cada distribuição e que proporcionou bons resultados para segmentação em imagens naturais (Rodrigues e Giraldi, 2009a). Dessa forma, neste trabalho será verificado se a extensão proposta por Rodrigues e Giraldi também oferece bons resultados como métrica de similaridade entre dois vetores para classificação de objetos sintéticos. Aplicando-se a generalização dos autores à divergência não-extensiva de Kullback-Leiber, tem-se a forma estendida da Equação (3.2):

$$Div_{KLq}(f \| f') = \sum_x f(x)^{q_A} \frac{1}{1 - q_A} (f(x)^{(1-q_A)} - f'(x)^{(1-q_B)}) \quad (3.2)$$

Onde  $q_B$  pode ser calculado em função de  $q_A$ :

$$q_B = \alpha q_A \quad (3.3)$$

## 3.4 Conclusões

Neste capítulo foram vistos os detalhes de como será feita a automação de processos por meio de reconhecimento de objetos sintéticos. Inicialmente foram descritas as principais características dos *clusters* nos quais os objetos de estudo serão classificados. Em seguida, foi apresentada uma visão geral do método proposto complementada por uma análise das técnicas de separação de objetos, classificação e reconhecimento que serão utilizadas.

A principal técnica estudada neste trabalho é a medida de similaridade entre dois vetores de características de objetos. Esta métrica é utilizada na classificação dos objetos de interesse permitindo a definição da maior similaridade entre o objeto a ser classificado e o centróide de cada classe.

# Capítulo 4

## RESULTADOS OBTIDOS

### 4.1 Metodologia Experimental

Para avaliar o método proposto, serão analisados os dois algoritmos que fazem uso de técnicas de visão computacional: a detecção de objetos gráficos e a classificação dos objetos extraídos. No primeiro algoritmo, o objetivo é detectar o objeto com o qual houve interação  $i$ , a partir da captura de tela e da coordenada do mouse em  $i$ . Por outro lado, para o segundo algoritmo, o objetivo é descobrir a classe da qual o objeto faz parte. Essa classificação é necessária para reduzir a quantidade de objetos candidatos no momento da execução, otimizando a velocidade do algoritmo de reprodução. Além disto, a classificação torna o reconhecimento mais robusto às variações que os objetos podem sofrer.

### 4.2 Extração de Objetos

Foram examinados quatro algoritmos diferentes de segmentação para fazer a detecção dos objetos: limiar adaptativo, entropia não-extensiva, entropia não-extensiva estendida e entropia não-extensiva estendida com os valores de  $q$  calculados automaticamente, também chamada de entropia não-extensiva automatizada. É interessante verificar nestes experimentos se a segmentação utilizando entropia não-extensiva pode ser uma técnica adequada para detectar objetos sintéticos em ambientes gráficos. Os experimentos realizados com limiar adaptativo servem de base de comparação, a partir do qual, os desempenhos dos outros algoritmos serão avaliados.

### 4.2.1 Critério de Avaliação

O método de detecção de objeto é executado para cada imagem  $P$  da base de capturas vista na Seção 3.3.2, junto com sua coordenada de interação  $c$ . O objeto resultante  $O$  é então comparado por meio de uma função de *matching* com o resultado esperado  $R$ . A detecção do objeto é dada como correta quando  $R$  é similar a  $O$ . A similaridade entre o  $R$  e  $O$  é dada pelo coeficiente de correlação  $C_C$  entre as duas imagens e pode ser calculado por meio da Equação (4.1).

$$C_C = \frac{\sum_{x,y} R_{x,y} O_{x,y} - \frac{1}{n} \left( \sum_{x,y} R_{x,y} \right) \left( \sum_{x,y} O_{x,y} \right)}{\sqrt{\sum_{x,y} R_{x,y}^2 - \frac{1}{n} \left( \sum_{x,y} R_{x,y} \right)^2} \sqrt{\sum_{x,y} O_{x,y}^2 - \frac{1}{n} \left( \sum_{x,y} O_{x,y} \right)^2}} \quad (4.1)$$

onde  $C_C$  é o coeficiente de correlação;  $O$  é o objeto extraído e  $R$  é o resultado esperado.

Para que as imagens seja dadas como similares, é necessário que este coeficiente seja maior que 84%, valor definido empiricamente durante a execução dos experimentos. Uma vez que para cálculo de  $C_C$  são necessárias duas imagens de mesmo tamanho, as figuras são redimensionadas para os maiores valores de largura e altura dos dois objetos. Ao final da execução do algoritmo para todas as imagens da base, a quantidade de objetos extraídos corretamente,  $E_{cor}$ , é contabilizada. O total de detecções corretas para cada algoritmo define o desempenho do método de detecção com aquele algoritmo de segmentação conforme a Equação (4.3).

$$D_{alg} = \frac{E_{cor}}{T} \quad (4.2)$$

onde  $D_{alg}$  é o desempenho do detector;  $E_{cor}$  é a quantidade de objetos extraídos corretamente conforme Equação (4.1) e  $T$  é o total de imagens da base.

### 4.2.2 Detecção Utilizando Limiar Adaptativo

O primeiro experimento realizado foi com o método de extração do objeto de interesse que utiliza limiar adaptativo como algoritmo segmentador. O método foi avaliado por meio de sua execução para extrair quinhentos objetos de cada classe; cada um a partir de um *screenshot* e uma coordenada onde houve o clique. Os resultados dos testes de desempenho com este algoritmo de extração servem de base de comparação com os resultados dos métodos de extração de objetos baseados em entropia que foram avaliados a seguir. Os resultados do primeiro experimento podem ser vistos no gráfico da Figura 4.1, onde o eixo Y representa a quantidade de objetos detectados

corretamente para cada uma das classes estudadas. Mais detalhes podem ser observados na Tabela 4.1, onde estão listadas as quantidades de objetos sintéticos detectados corretamente e o desempenho do algoritmo para cada classe.

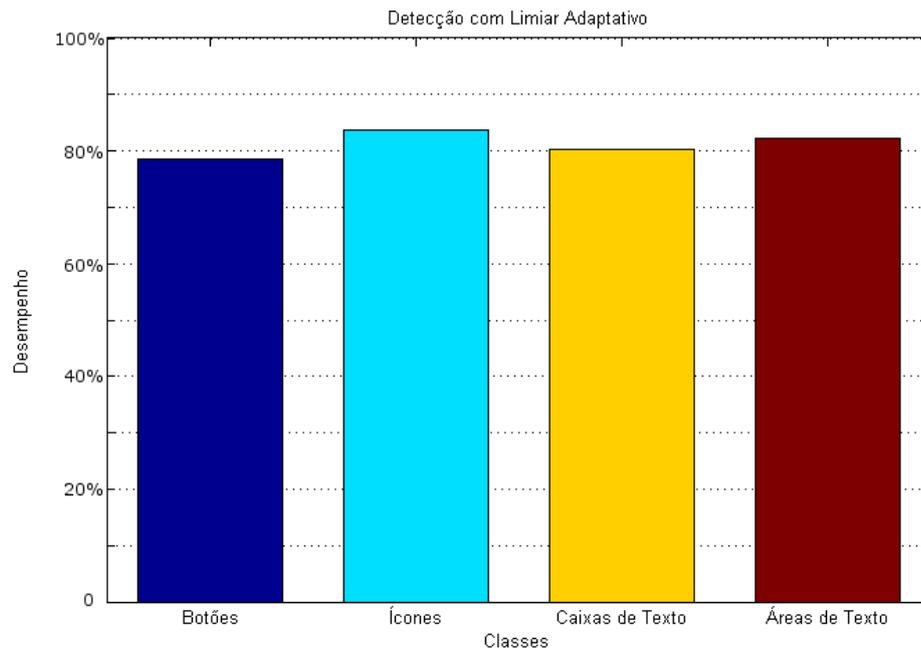


Figura 4.1: Detecção de objetos utilizando limiar adaptativo como algoritmo de segmentação

Tabela 4.1: Resultados da detecção de objetos utilizando limiar adaptativo como algoritmo de segmentação

Classe	Elementos Avaliados	Extrações Corretas	Desempenho
Botões	500	393	78,6%
Ícones	500	418	83,6%
Áreas de Texto	500	411	82,2%
Caixas de Texto	500	401	80,2%

Este algoritmo segmentador se mostrou bastante eficiente para detecção de objetos, uma vez que a extração foi correta em mais de 78% em todas as classes. Como pode-se observar na Tabela 4.1, o algoritmo de extração de objetos conseguiu extrair uma quantidade maior de ícones, caixas de texto e áreas de texto corretamente. Isto se deve ao fato de que estes elementos possuem bordas bastante diferentes do *background*, facilitando a extração destes elementos.

### 4.2.3 Detecção Utilizando Entropia de Tsallis

O segundo experimento foi realizado utilizando a técnica de segmentação com entropia de Tsallis como método para detectar objetos. Conforme já visto na Seção 2.3.1, a entropia não-extensiva  $S_q(p)$  de uma distribuição  $p$  é calculada utilizando-se a Equação (2.6), onde o valor de  $q$  influencia diretamente sobre o valor de  $S_q(p)$ , proporcionando diferentes resultados a medida em que o valor de  $q$  varia. Portanto, para avaliar este método, foi necessária a execução do algoritmo de detecção com valores diferentes de  $q$ , variando no intervalo entre  $q = 0,05$  até  $q = 2,00$ . No gráfico da Figura 4.2, o eixo Y representa o desempenho do algoritmo de segmentação por meio de entropia não-extensiva para cada valor de  $q$ . Na Tabela 4.2, estão listadas as quantidades de detecções realizadas corretamente por classe e para cada valor de  $q$ .

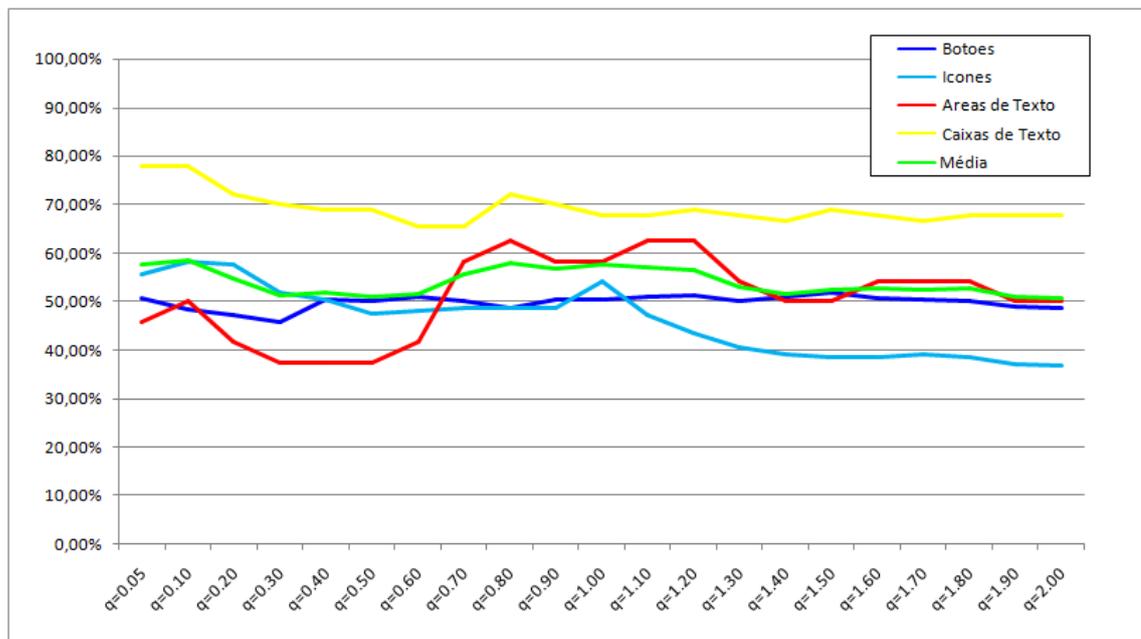


Figura 4.2: Resultados da detecção de objetos utilizando entropia não-extensiva como algoritmo segmentador variando o valor de  $q$

Tabela 4.2: Resultados da detecção de objetos utilizando entropia não-extensiva como algoritmo segmentador variando o valor de  $q$

Valor de $q$	Botões	Ícones	Áreas de Texto	Caixas de Texto	Média
$q=0,05$	50,69%	55,71%	45,83%	77,78%	57,50%
$q=0,10$	48,28%	58,10%	50,00%	77,78%	58,54%
$q=0,20$	47,24%	57,62%	41,67%	72,22%	54,69%

Tabela 4.2 – continuação da página anterior

Valor de $q$	Botões	Ícones	Áreas de Texto	Caixas de Texto	Média
$q=0,30$	45,86%	51,90%	37,50%	70,00%	51,32%
$q=0,40$	50,34%	50,48%	37,50%	68,89%	51,80%
$q=0,50$	50,00%	47,62%	37,50%	68,89%	51,00%
$q=0,60$	51,03%	48,10%	41,67%	65,56%	51,59%
$q=0,70$	50,00%	48,57%	58,33%	65,56%	55,62%
$q=0,80$	48,62%	48,57%	62,50%	72,22%	57,98%
$q=0,90$	50,34%	48,57%	58,33%	70,00%	56,81%
$q=1,00$	50,34%	54,29%	58,33%	67,78%	57,69%
$q=1,10$	51,03%	47,14%	62,50%	67,78%	57,11%
$q=1,20$	51,38%	43,33%	62,50%	68,89%	56,53%
$q=1,30$	50,00%	40,48%	54,17%	67,78%	53,11%
$q=1,40$	51,03%	39,05%	50,00%	66,67%	51,69%
$q=1,50$	51,72%	38,57%	50,00%	68,89%	52,30%
$q=1,60$	50,69%	38,57%	54,17%	67,78%	52,80%
$q=1,70$	50,34%	39,05%	54,17%	66,67%	52,56%
$q=1,80$	50,00%	38,57%	54,17%	67,78%	52,63%
$q=1,90$	48,97%	37,14%	50,00%	67,78%	50,97%
$q=2,00$	48,62%	36,67%	50,00%	67,78%	50,77%

No gráfico da Figura 4.2 e na Tabela 4.2 pode-se observar que o desempenho do detector de objetos está diretamente relacionado com o valor de  $q$  utilizado pelo algoritmo de segmentação. Além disso, observando as colunas 3,5 e 6 da Tabela 4.2, nota-se que o melhor resultado geral foi obtido com o valor de  $q = 0,1$ , que proporcionou um desempenho médio de 58,54% e também proporcionou a melhor detecção de ícones e caixas de texto para este algoritmo. Na coluna 2 da Tabela 4.2, é possível verificar que o melhor desempenho para detecção de botões foi obtido para  $q = 1,5$ . A coluna 4 da Tabela 4.2, mostra que os melhores desempenhos para detecção de áreas de texto foram obtidos para  $q = 0,80; q = 1,10; q = 1,20$ .

#### 4.2.4 Detecção Utilizando Q-Entropia Estendida

O terceiro experimento foi realizado utilizando a forma estendida de q-entropia proposta por Rodrigues e Giraldi como algoritmo segmentador (Rodrigues e Giraldi, 2009a). Neste caso, conforme já visto na Seção 2.3.1, o resultado da segmentação depende dos parâmetros  $q_1$  e  $\alpha$ . Esses valores podem ser obtidos empiricamente por meio do cálculo da entropia não-extensiva para  $q_1$  e  $\alpha$  com valores de  $q_1$  variando entre 0,01 e 2 e  $\alpha$  variando entre 0,5 e 5, para todas as imagens da Base A, verificando para qual par de valores de  $q_1$  e  $\alpha$  é obtida a maior quantidade de objetos sintéticos detectados corretamente. No gráfico da Figura 4.3, o eixo Y representa o desempenho do algoritmo de segmentação por meio do método de Rodrigues e Giraldi para cada valor de  $q$ . Na Tabela 4.3, estão listadas as quantidades de detecções realizadas corretamente por classe para quatro combinações de  $\alpha$  e  $q$ . Os dados completos obtidos durante a execução deste experimento podem ser vistos com mais detalhe na Tabela A.1 do Apêndice A.

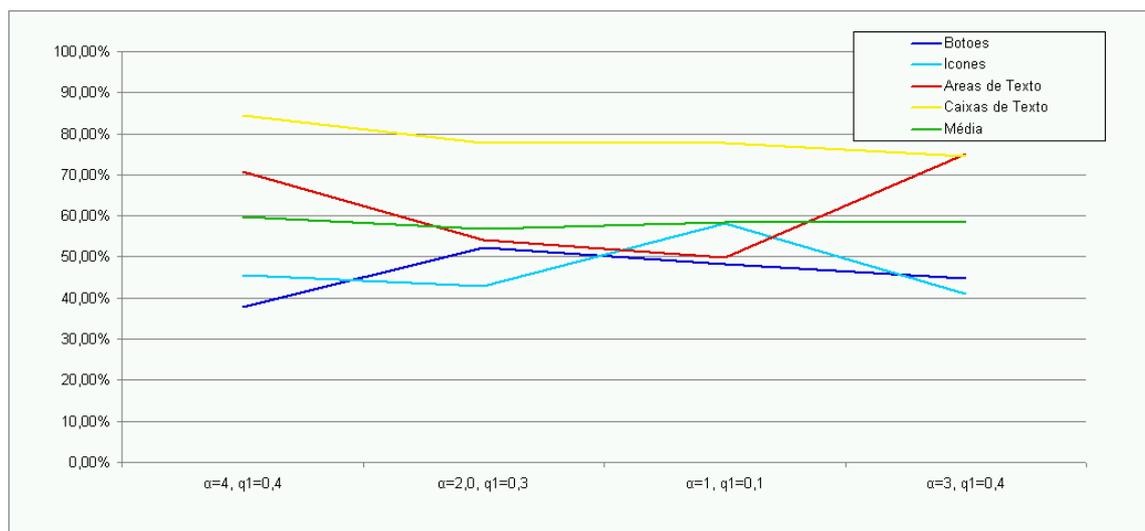


Figura 4.3: Resultados da detecção de objetos utilizando entropia não-extensiva como algoritmo segmentador variando o valor de  $\alpha$  e  $q_1$

Tabela 4.3: Resumo dos resultados da detecção de objetos utilizando q-entropia estendida como algoritmo segmentador variando o valor de  $\alpha$  e  $q$ . Os dados completos podem ser encontrados na Tabela A.1 do Apêndice A.

Valor de $\alpha$	Valor de $q_1$	Botões	Ícones	Áreas de Texto	Caixas de Texto	Média
4,0	0,4	37,93%	45,71%	70,83%	84,44%	59,72%
2,0	0,3	52,41%	42,86%	54,17%	77,78%	56,80%
1,0	0,1	48,28%	58,1%	50,00%	77,78%	58,54%
3,0	0,4	44,83%	40,95%	75,00%	74,44%	58,80%

Como pode-se observar na coluna 7 da Tabela 4.3, o melhor do desempenho geral foi obtido para  $\alpha = 4,0$  e  $q_1 = 0,4$  que também propiciou a melhor detecção de caixas de texto. Na coluna 3, verifica-se que o melhor desempenho para segmentação de botões foi obtido para os valores  $\alpha = 2,0$  e  $q_1 = 0,3$ . A coluna 4 nos mostra que o melhor desempenho para detecção de ícones foi obtido para  $\alpha = 1,0$  e  $q_1 = 0,1$ . A melhor segmentação para detecção de áreas de texto foi obtida com valores  $\alpha = 3,0$  e  $q_1 = 0,4$ , como observa-se na coluna 5 da Tabela 4.3.

#### 4.2.5 Detecção Utilizando Q-Entropia Automatizada

A definição do valor ótimo de  $q$  é um dos problemas encontrados quando é utilizada a entropia de Tsallis ou sua forma estendida. Na Seção 2.3.1, foi visto que Rodrigues e Giraldi propuseram um método para cálculo automático deste parâmetro (Rodrigues e Giraldi, 2009b). No método proposto, calcula-se a relação entre a entropia não-extensiva e a entropia máxima de uma distribuição para cada valor de  $q$ , variando entre 0,1 até 2,0. Com o valor de  $q_1$  calculado automaticamente, calcula-se o valor de  $q_2$  em função de  $\alpha$ . Dessa forma, este experimento foi conduzido variando o valor de  $\alpha$  entre 0,1 e 9,0. Os resultados podem ser vistos no gráfico da Figura 4.4, onde o eixo Y representa o desempenho do algoritmo de segmentação por meio do método automatizado de Rodrigues e Giraldi para cada valor de  $\alpha$ . Na Tabela 4.4, estão listados os desempenhos obtidos para cada valor de  $\alpha$  para cada classe estudada.

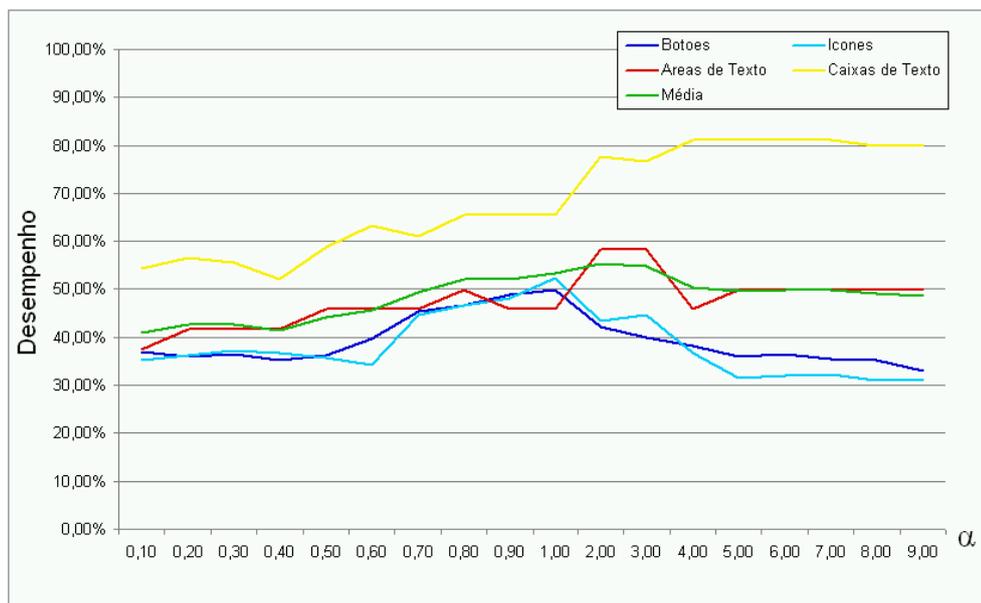


Figura 4.4: Resultados da detecção de objetos utilizando q-entropia automatizada como algoritmo segmentador variando o valor de  $\alpha$

Tabela 4.4: Resultados da detecção de objetos utilizando q-entropia automatizada como algoritmo segmentador variando o valor de  $\alpha$

Valor de $\alpha$	Botões	Ícones	Áreas de Texto	Caixas de Texto	Média
$\alpha=0,10$	36,90%	35,24%	37,50%	54,44%	41,02%
$\alpha=0,20$	35,86%	36,19%	41,67%	56,67%	42,60%
$\alpha=0,30$	36,55%	37,14%	41,67%	55,56%	42,73%
$\alpha=0,40$	35,17%	36,67%	41,67%	52,22%	41,43%
$\alpha=0,50$	36,21%	35,71%	45,83%	58,89%	44,16%
$\alpha=0,60$	39,66%	34,29%	45,83%	63,33%	45,78%
$\alpha=0,70$	45,52%	44,76%	45,83%	61,11%	49,31%
$\alpha=0,80$	46,55%	46,67%	50,00%	65,56%	52,19%
$\alpha=0,90$	48,97%	48,10%	45,83%	65,56%	52,11%
$\alpha=1,00$	50,00%	52,38%	45,83%	65,56%	53,44%
$\alpha=2,00$	42,07%	43,33%	58,33%	77,78%	55,38%
$\alpha=3,00$	40,00%	44,76%	58,33%	76,67%	54,94%
$\alpha=4,00$	38,28%	36,67%	45,83%	81,11%	50,47%
$\alpha=5,00$	35,86%	31,43%	50,00%	81,11%	49,60%
$\alpha=6,00$	36,55%	31,90%	50,00%	81,11%	49,89%
$\alpha=7,00$	35,52%	32,38%	50,00%	81,11%	49,75%
$\alpha=8,00$	35,17%	30,95%	50,00%	80,00%	49,03%
$\alpha=9,00$	33,10%	30,95%	50,00%	80,00%	48,51%

No gráfico da Figura 4.4 e na Tabela 4.4 pode-se observar que o desempenho do detector de objetos está diretamente relacionado com o valor de  $\alpha$  utilizado pelo algoritmo de segmentação. Além disso, observando as colunas 6 da Tabela 4.4, nota-se que o melhor desempenho geral foi obtido com o valor de  $\alpha = 2,0$ , que proporcionou um desempenho médio de 55,38%. No entanto, as colunas 2 e 3 da Tabela 4.4, mostram que os melhores desempenhos para a detecção de botões e ícones foi obtido para  $\alpha = 1,0$ . A coluna 4 da Tabela 4.4, mostra que o melhor desempenho para detecção de áreas de texto foi obtido para  $\alpha = 3,0$ , enquanto que a coluna 5 demonstra que o melhor desempenho na detecção de caixas de texto foi obtido para  $\alpha = 4; \alpha = 5; \alpha = 6; \alpha = 7$ .

## 4.2.6 Comparação dos Algoritmos de Segmentação

Após a execução dos experimentos com os métodos de detecção dos objetos, é possível fazer a comparação dos resultados obtidos. Os desempenhos dos algoritmos de segmentação podem ser vistos no gráfico da Figura 4.5, onde o eixo Y representa o desempenho de cada método de detecção de objetos sintéticos utilizado. Na Tabela 4.5, estão listados os desempenhos obtidos para cada algoritmo e para cada classe estudada.

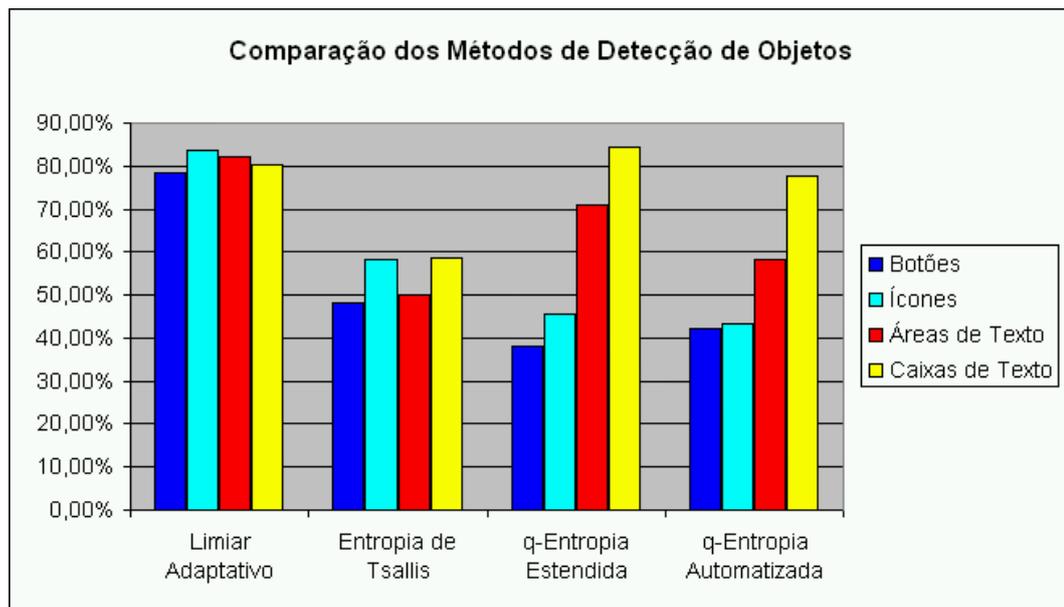


Figura 4.5: Comparação dos desempenhos dos algoritmos utilizados para detecção de objetos sintéticos

Tabela 4.5: Detalhe da comparação dos desempenhos dos algoritmos utilizados para detecção de objetos sintéticos

Algoritmo	Bot.	Ico.	ATx.	Cxs.	Obs.
Limiar Adaptativo	78,60%	83,60%	82,20%	80,20%	
Entropia de Tsallis	48,28%	58,10%	50,00%	58,54%	$q = 0,1$
q-Entropia Estendida	37,93%	45,71%	70,83%	84,44%	$\alpha=4$ e $q1=0,4$
q-Entropia Automática	42,07%	43,33%	58,33%	77,78%	$\alpha=2$

A partir dos resultados obtidos, nota-se que, nas condições estudadas, a detecção de objetos utilizando limiar adaptativo como algoritmo segmentador oferece melhor desempenho em relação aos outros métodos estudados para todas as classes, com exceção das caixas de texto, classe para a qual, o melhor desempenho foi obtido com a forma estendida da segmentação por entropia não-extensiva.

## 4.3 Atributos dos Objetos

Para realizar a classificação de objetos, é necessária a caracterização do mesmo por meio de atributos que permitam representá-los de uma forma que possibilite a diferenciação dos demais. Assim, foram escolhidos seis atributos para serem extraídos de cada objeto a fim de classificá-lo em uma das quatro classes estudadas.

### 4.3.1 Histograma de Cores

O primeiro atributo a ser extraído dos objetos é o histograma de cores. Neste trabalho, foi utilizado o espaço de cores HSV com intensidade de dezoito valores para matiz (H), três valores para saturação (S) e três valores para intensidade (V). Assim, o histograma de uma imagem neste espaço possui 162 estados, cada um deles representando uma combinação dos valores de H, S e V. O gráfico da Figura 4.6 representa o comportamento da média dos histogramas para cada classe. O eixo Y representa a intensidade com que cada combinação de valores ocorre em cada classe.

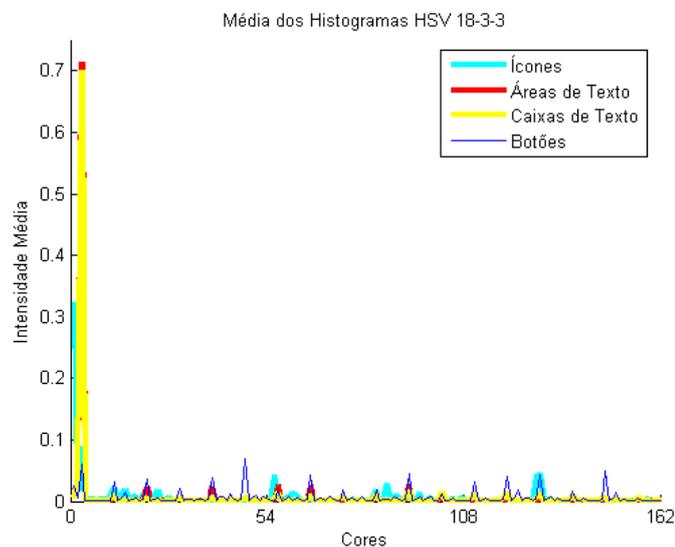


Figura 4.6: Atributo F1 - Histograma HSV 18-3-3

Pode-se observar no gráfico da Figura 4.6 que as classes de Botões e Ícones apresentam padrões mais distribuídos de cores, uma vez que esses objetos possuem grande diversidade de cores enquanto que as classes Áreas de Texto e Caixas de Texto possuem uma grande quantidade de pixels com cor branca, representada pela posição 2 no eixo X.

### 4.3.2 Variação de Intensidade Média Vertical

O segundo atributo a ser extraído dos objetos é a variação de intensidade média vertical. Para extração deste atributo, são consideradas os valores médios da intensidade de tons de cinza ao longo do eixo Y, em 30 amostragens, como observa-se na Figura 4.7. O Gráfico da Figura 4.8, representa o comportamento da média dos histogramas para cada classe. O eixo Y representa a intensidade de tons de cinza para cada amostragem do eixo X.



Figura 4.7: Extração do atributo F2 - Variação de Intensidade Média Vertical

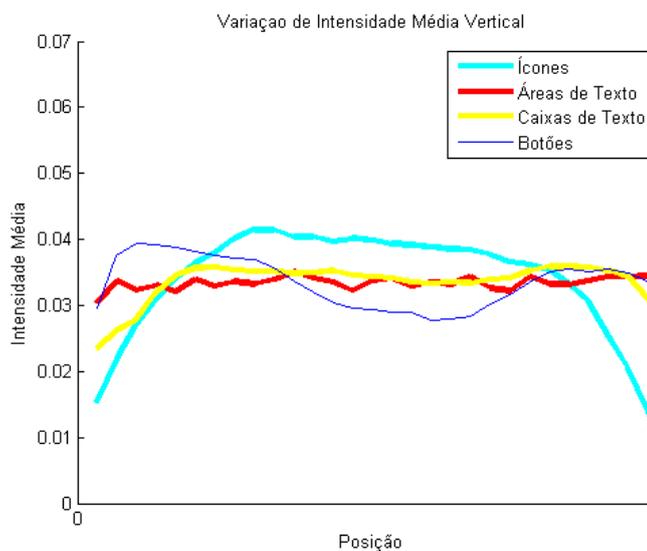


Figura 4.8: Atributo F2 - Variação de Intensidade Média Vertical

Como pode-se observar na Figura 4.8, o comportamento dessa característica para ícones é bem diferente dos demais. Além disso, observa-se que o comportamento desse atributo para botões é mais discriminativo em relação a caixas de texto do que em relação a áreas de texto. Nota-se também que as maiores diferenças entre as curvas de cada objeto estão nas áreas periféricas do gráfico, para os menores e maiores valores de x.

### 4.3.3 Histograma de Raios

O terceiro atributo a ser extraído dos objetos é o histograma de raios. Para extração deste atributo, são medidas 10 distancias entre o centro do objeto e suas fronteiras, como pode-se observar na Figura 4.9. O Gráfico da Figura 4.12, representa o comportamento da média dos histogramas para cada classe. O eixo Y representa o tamanho relativo de cada raio.

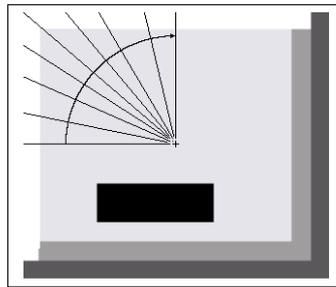


Figura 4.9: Extração do atributo F3 - Histograma de Raios

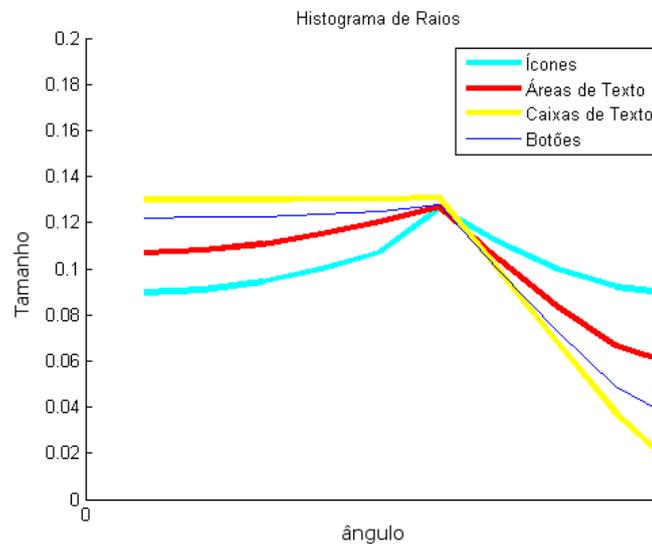


Figura 4.10: Atributo F3 - Histograma de Raios

A Figura 4.10 demonstra o comportamento dessa característica para caixas de texto é bem diferente dos demais, particularmente da curva de ícones onde a diferença é maior. Além disso, observa-se que o comportamento desse atributo para botões é mais discriminativo em relação a áreas de texto do que em relação a ícones.

### 4.3.4 Variação Vertical de Intensidades

O quarto atributo a ser extraído dos objetos é a variação vertical de intensidades de tons de cinza. Para a extração deste atributo, é realizada uma varredura vertical no centro objeto, medindo-se as intensidades de tons-de cinza em 30 amostras ao longo da extensão vertical do objeto, como pode-se observar na Figura 4.11. O Gráfico da Figura 4.12, representa o comportamento da média das histogramas para cada classe. O eixo Y representa a intensidade de cada amostra do objeto.

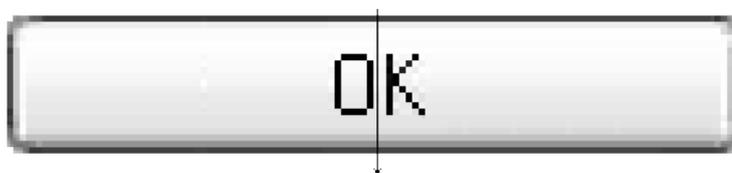


Figura 4.11: Extração do atributo F4 - Variação vertical de intensidades de tons de cinza

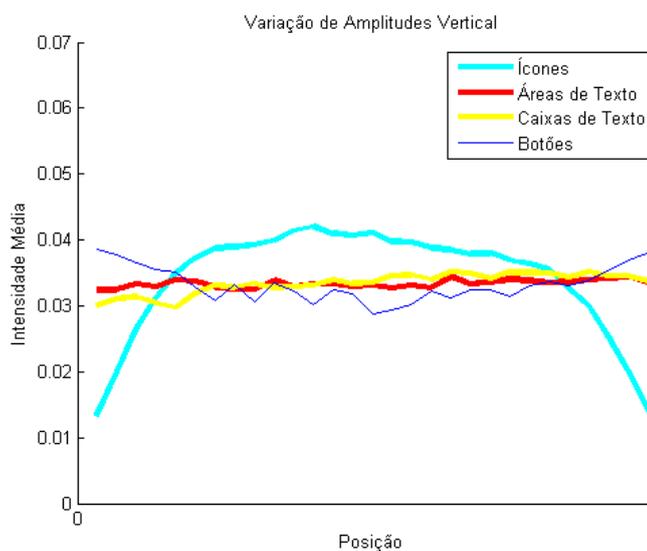


Figura 4.12: Atributo F4 - Variação Vertical de Intensidades de Cinza

Pode-se observar na Figura 4.12 que o comportamento dessa característica para ícones e botões é bem diferente dos demais. Além disso, observa-se que o comportamento desse atributo é pouco discriminativo em relação a caixas e áreas de texto. Nota-se também que as maiores diferenças entre as curvas de cada objeto estão nas áreas periféricas do gráfico, para os menores e maiores valores de x.

### 4.3.5 Variação Horizontal de Intensidades

O quinto atributo a ser extraído dos objetos é a variação horizontal de intensidades de tons de cinza. Para a extração deste atributo, é realizada uma varredura horizontal no centro objeto, medindo-se as intensidades de tons-de cinza em 30 amostras ao longo da extensão horizontal do objeto, como pode-se observar na Figura 4.13. O Gráfico da Figura 4.14, representa o comportamento da média das histogramas para cada classe. O eixo Y representa a intensidade de cada amostra do objeto.



Figura 4.13: Extração do atributo F5 - Variação horizontal de intensidades de tons de cinza

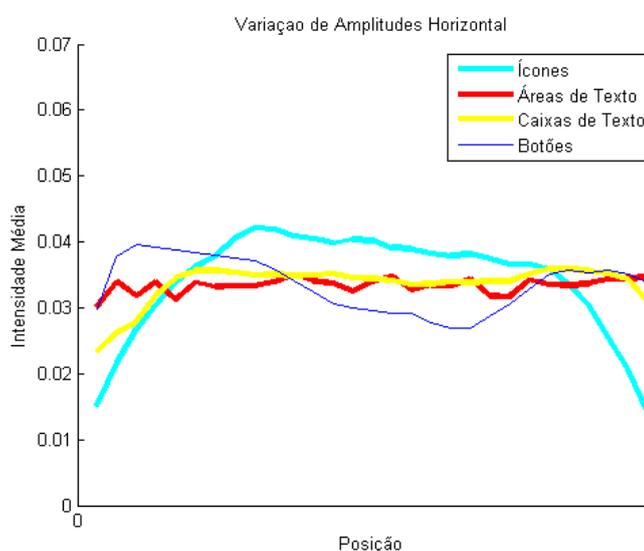


Figura 4.14: Atributo F5 - Variação horizontal de de intensidades de tons de cinza

Como observa-se na Figura 4.14, o comportamento dessa característica varia bastante entre as classes, fazendo com que este atributo tenha bastante potencial discriminante. Além disso, observa-se que o comportamento desse atributo é mais discriminativo em relação a caixas e áreas de texto do que o quarto atributo. Nota-se também que as maiores diferenças entre as curvas de cada objeto estão nas áreas periféricas do gráfico, para os menores e maiores valores de x.

### 4.3.6 Variação Diagonal de Intensidades

O sexto atributo a ser extraído dos objetos é a variação diagonal de intensidades de tons de cinza. Para a extração deste atributo, é realizada uma varredura diagonal passando pelo centro objeto, medindo-se as intensidades de tons-de cinza em 30 amostras ao longo do corte diagonal do objeto, como pode-se observar na Figura 4.15. O Gráfico da Figura 4.16, representa o comportamento da média das histogramas para cada classe. O eixo Y representa a intensidade de cada amostra do objeto.



Figura 4.15: Extração do atributo F6 - Variação diagonal de intensidades de tons de cinza

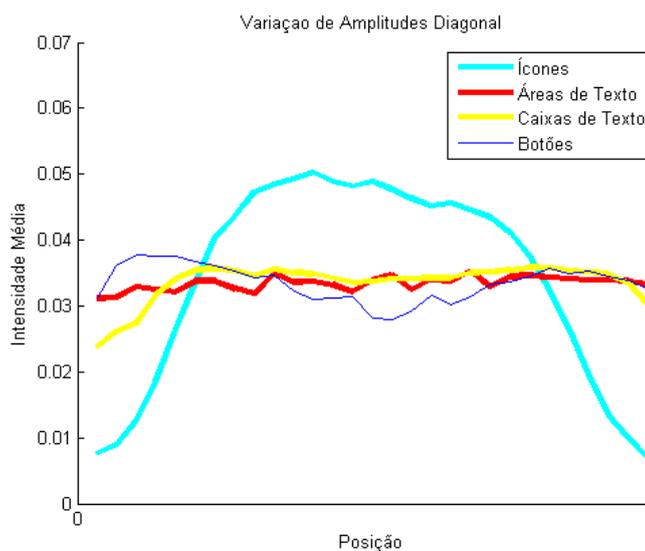


Figura 4.16: Atributo F6 - Variação diagonal de intensidades de tons de cinza

A Figura 4.16 mostra o que esta característica varia pouco entre todas as classes com exceção da curva de ícones, a qual é bastante diferenciada das demais. Assim, pode-se dizer que este atributo é bastante discriminativo para ícones, e pouco discriminativo para as demais classes.

## 4.4 Classificação de Imagens

Foram examinados cinco métodos diferentes para fazer a classificação dos objetos: distância euclidiana, e distância de Kullback-Leiber em suas variações clássica, não-extensiva, estendida e com os valores de  $q$  calculados automaticamente, também chamada de distância de Kullback-Leiber automatizada. A principal finalidade destes experimentos é verificar se o uso da distância de Kullback-Leiber como métrica de similaridade entre vetores propicia uma boa classificação de objetos sintéticos. O experimento realizado com distância euclidiana serve de base de comparação, a partir da qual, os desempenhos dos outros algoritmos serão avaliados.

De cada objeto da Base de Capturas, foram extraídos seis atributos em forma de distribuições probabilísticas. A seleção dessas características foi feita procurando utilizar os atributos mais significativos de cada classe. O desempenho dos algoritmos de classificação depende em grande parte das características extraídas dos objetos. Enquanto que características que variam pouco entre as classes tornam a classificação mais difícil de ser realizada, características com alta capacidade discriminativa fazem com que a classificação atinja altos níveis de desempenho.

### 4.4.1 Critério de Avaliação

O método de classificação de objetos é executado para cada imagem da base B, vista na Seção 3.3.2. O resultado do algoritmo de agrupamento é dado como correto quando a classe em que o objeto é classificado é a mesma na qual o objeto foi rotulado inicialmente. Com cada um dos algoritmos analisados são executadas as classificações de todos os objetos da base de dados resultando no total de classificações corretas para cada classe em cada experimento. Ao final da execução do algoritmo, a quantidade de objetos classificados corretamente,  $C_{cor}$ , é contabilizada. O total de classificações corretas para cada algoritmo define o desempenho do método de agrupamento conforme a Equação (4.3).

$$D_{class} = \frac{C_{cor}}{T} \quad (4.3)$$

onde  $D_{class}$  é o desempenho do classificador;  $C_{cor}$  é a quantidade de objetos classificados corretamente e  $T$  é o total de imagens da base.

#### 4.4.2 Classificação com distância Euclidiana

No primeiro experimento executado, foi utilizada a distância euclidiana como métrica de similaridade entre os centróides de cada classe e os atributos extraídos de cada objeto para sua classificação. O desempenho do algoritmo pode ser observado no gráfico da Figura 4.17, onde o eixo Y representa o desempenho do método para cada uma das classes estudadas. Os detalhes dos resultados do experimento estão listados na Tabela 4.6, onde em cada linha estão as contagens dos objetos classificados como pertencentes a cada classe, por exemplo: Na primeira linha, dentre todos os 976 objetos rotulados como botões, 621 foram classificados corretamente, enquanto que 8 foram classificados como ícones, 150 como áreas de texto e 197 como caixas de texto, somando um total de 355 falsos negativos listados na sétima coluna da Tabela 4.6. O total de falsos positivos listados na sexta coluna correspondem à quantidade de objetos de outras classes que foram classificados como botões.

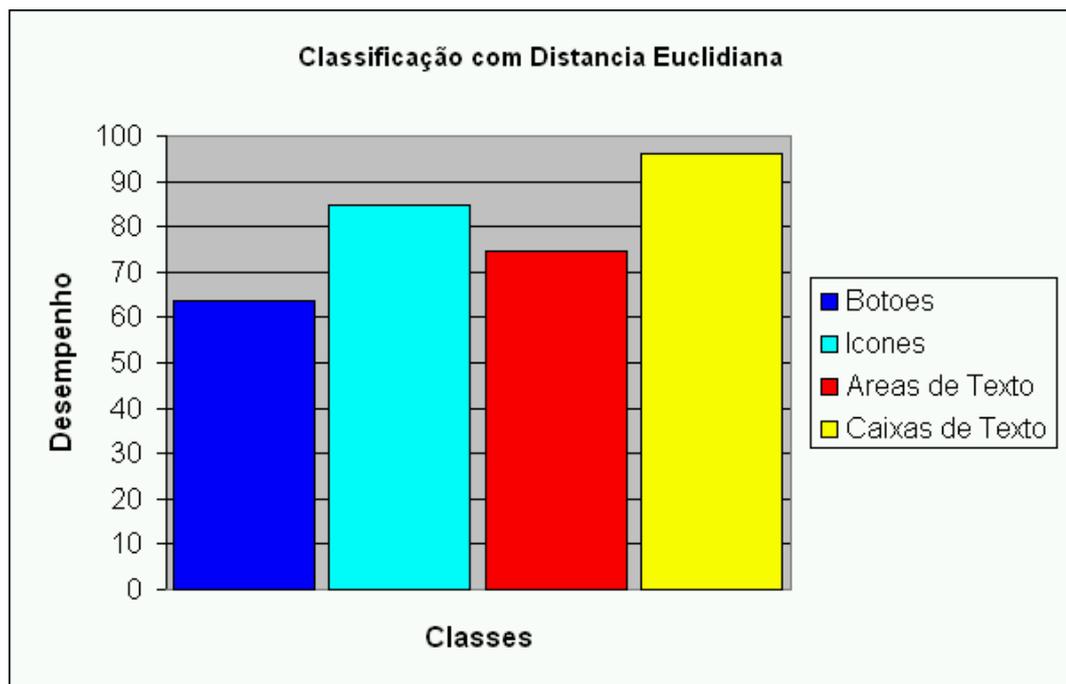


Figura 4.17: Resultados da classificação de objetos utilizando distância euclidiana como métrica de similaridade

Tabela 4.6: Resultados da classificação de objetos utilizando distância euclidiana como métrica de similaridade

Classe	Qtd. Bot.	Qtd. Ico.	Qtd. ATx.	Qtd. Cxs.	Desempenho	FP	FN
Botões	621	8	150	197	63,62%	59	355
Ícones	37	943	128	2	84,95%	8	167
Áreas de Texto	16	0	298	86	74,50%	292	102
Caixas de Texto	6	0	14	515	96,26%	285	20

Os melhores desempenhos obtidos com este algoritmo foram as classificações de caixas de texto e ícones, respectivamente com 96,26% e 84,95% enquanto que o menor desempenho foi o da classificação de botões. No entanto, todas os objetos foram classificados corretamente com desempenho maior que 63% nas quatro classes estudadas.

#### 4.4.3 Classificação com distância de Kullback-Leiber Clássica

O segundo experimento foi executado utilizando a distância clássica de Kullback-Leiber, também conhecida como entropia como medida de similaridade para classificação. Como já foi dito na Seção 2.4.3, a divergência de Kullback-Leiber não pode ser considerada uma distância, uma vez que, para duas distribuições probabilísticas  $d_1$  e  $d_2$ ,  $Div_{KL}(d_1||d_2) \neq Div_{KL}(d_2||d_1)$ . Sendo assim, o cálculo da distância de Kullback-Leiber é feito com a soma ponderada das divergências.

O desempenho do algoritmo pode ser observado no gráfico da Figura 4.18, onde o eixo Y representa o desempenho do método para cada uma das classes estudadas. Os detalhes dos resultados do experimento estão listados na Tabela 4.7, onde em cada linha estão as contagens dos objetos classificados como pertencentes a cada classe.

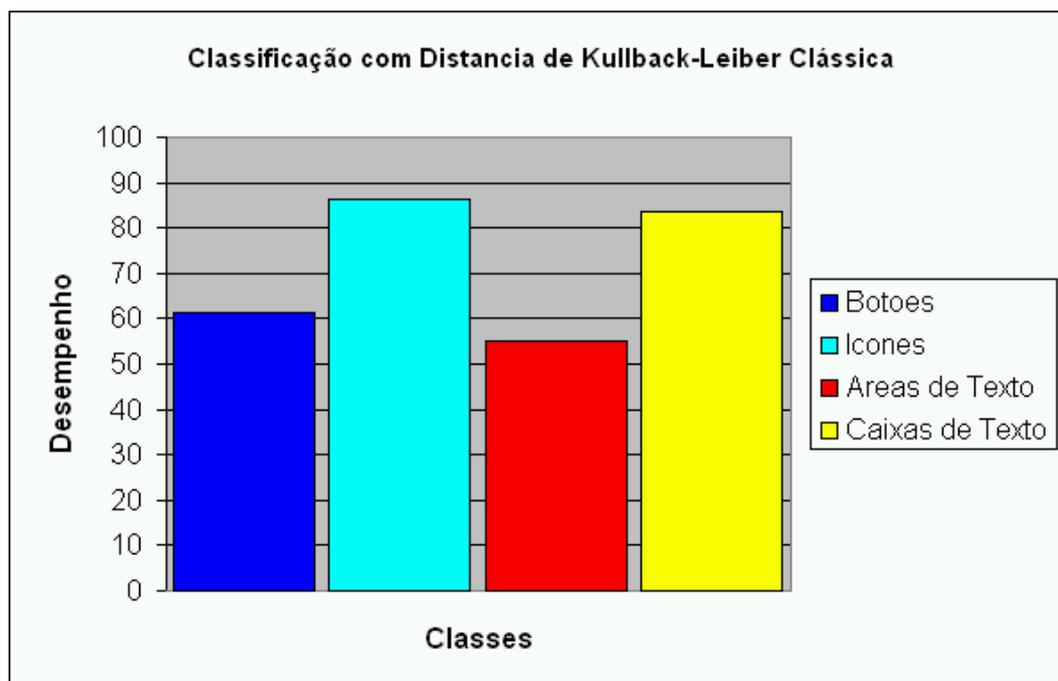


Figura 4.18: Resultados da classificação de objetos utilizando distância clássica de Kullback-Leiber como métrica de similaridade

Tabela 4.7: Resultados da classificação de objetos utilizando distância clássica de Kullback-Leiber como métrica de similaridade

Classe	Qtd. Bot.	Qtd. Ico.	Qtd. ATx.	Qtd. Cxs.	Desempenho	FP	FN
Botões	597	103	92	184	61,16%	164	379
Ícones	106	959	30	15	86,39%	106	151
Áreas de Texto	28	0	220	152	55,00%	176	180
Caixas de Texto	30	3	54	448	83,73%	351	87

Os melhores desempenhos obtidos com este algoritmo foram as classificações de ícones e caixas de texto, respectivamente com 86,39% e 83,73% enquanto que o menor desempenho foi o da classificação de áreas de texto. No entanto, todas os objetos foram classificados corretamente com desempenho maior que 50% nas quatro classes estudadas.

#### 4.4.4 Classificação com Distância de Kullback-Leiber Não-Extensiva

No experimento de número três, foi utilizada a distância não-extensiva de Kullback-Leiber, como métrica de similaridade para classificação. O resultado obtido com esta técnica depende do valor de  $q$  utilizado. Assim, este algoritmo foi executado para diferentes valores de  $q$ , variando no intervalo entre 0,1 e 2. O desempenho do algoritmo para cada classe em função de  $q$  pode ser observado no gráfico da Figura 4.19, onde o eixo Y representa o desempenho do método para cada uma das classes estudadas. Os detalhes dos resultados do experimento estão listados na Tabela 4.8, onde em cada linha estão os desempenhos obtidos com a classificação dos objetos rotulados em cada classe. Por exemplo: Na primeira linha, foi utilizado o valor de  $q = 0,1$  e, entre todos os botões da base de objetos, 70,59% foram classificados corretamente. Da mesma forma, entre todos os ícones, 85,68% foram classificados corretamente.

Tabela 4.8: Resultados da classificação de objetos utilizando distância não-extensiva de Kullback-Leiber como métrica de similaridade

$q$	Des. Botões	Des. Ícones	Des. Áreas de Texto	Des. Caixas de Texto
0,1	70,59%	85,68%	74,25%	96,45%
0,2	68,85%	85,50%	74,00%	97,38%
0,3	65,98%	85,23%	74,75%	97,38%

Tabela 4.8 – continuação da página anterior

$q$	Des. Botões	Des. Ícones	Des. Áreas de Texto	Des. Caixas de Texto
0,4	64,75%	85,05%	74,25%	97,38%
0,5	63,63%	84,95%	74,50%	96,26%
0,6	55,84%	84,59%	74,00%	95,33%
0,7	48,46%	84,14%	71,25%	94,21%
0,8	38,63%	84,50%	67,00%	89,91%
0,9	28,38%	82,70%	63,75%	77,01%
1,0	61,17%	86,40%	55,00%	83,74%
1,1	34,63%	86,31%	53,50%	71,59%
1,2	48,36%	86,76%	48,00%	73,46%
1,3	49,08%	87,12%	47,25%	70,09%
1,4	47,44%	86,40%	45,00%	67,48%
1,5	45,80%	85,50%	43,50%	64,67%
1,6	44,57%	84,77%	41,75%	61,87%
1,7	44,26%	84,23%	41,25%	60,19%
1,8	43,85%	83,69%	41,00%	58,88%
1,9	42,93%	83,06%	40,00%	56,07%
2,0	42,83%	82,52%	38,75%	53,64%

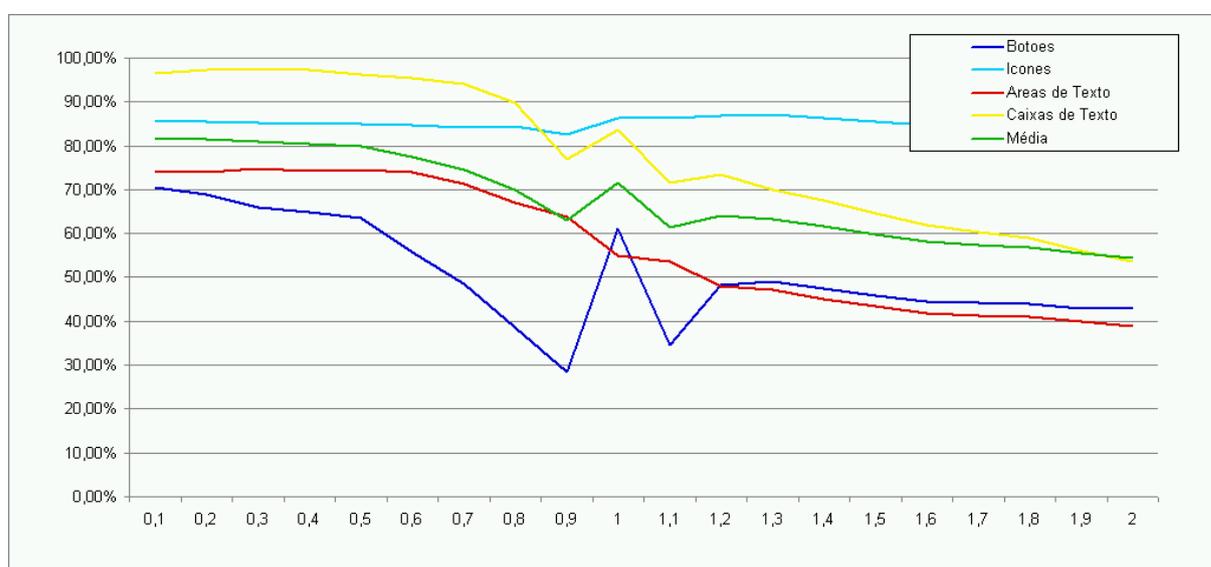


Figura 4.19: Resultados da classificação de objetos utilizando distância não-extensiva de Kullback-Leiber como métrica de similaridade

Na Tabela 4.8 e no gráfico da Figura 4.19, é possível observar que o melhor desempenho deste método foi obtido com o valor de  $q = 0,1$  (primeira linha da Tabela 4.8) e que o desempenho do algoritmo foi bastante satisfatório para valores de  $q$  inferiores a 0,5. Além disso, a coluna 3 da Tabela 4.8 mostra que para a  $q = 1,3$  foi obtido o maior quantidade de ícones classificados corretamente. Da mesma maneira, as demais colunas da Tabela 4.8 mostram que os melhores desempenhos para as classes de botões, áreas e caixas de texto foram obtidos para o primeiro valor de  $q$  utilizado, ou seja,  $q = 01$ .

#### 4.4.5 Classificação com q-Distância Estendida de Kullback-Leiber

No quarto experimento, a classificação é feita utilizando-se a forma estendida da  $q$ -distância de Kullback-Leiber proposta por Rodrigues e Giraldi conforme a Equação (3.2). O resultado obtido com esta técnica depende dos valores de  $q_1$  e  $\alpha$  utilizados. Dessa forma, este algoritmo foi executado para diferentes combinações desses valores, variando  $q_1$  no intervalo entre 0,01 e 2 e  $\alpha$  no intervalo entre 0,5 e 9. Os dados completos obtidos com a execução deste experimento estão listados na Tabela B.1 do Apêndice B. No gráfico da Figura 4.20, estão alguns dos resultados obtidos na execução deste experimento, onde o eixo Y representa do desempenho do algoritmo para cada valor de  $\alpha$  e  $q_1$ .

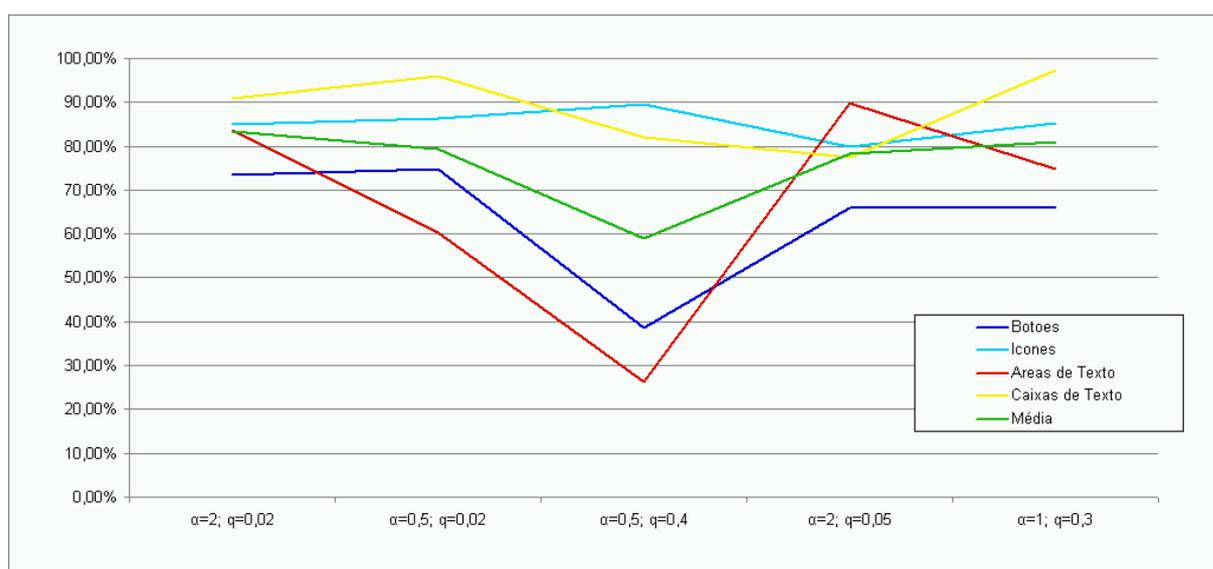


Figura 4.20: Resultados da classificação de objetos utilizando  $q$ -distância estendida de Kullback-Leiber como métrica de similaridade.

Tabela 4.9: Resultados da classificação de objetos utilizando  $q$ -distância estendida de Kullback-Leiber como métrica de similaridade para  $\alpha=2$  e  $q_1=0,02$

Valor de $\alpha$	Valor de $q_1$	Botões	Ícones	Áreas de Texto	Caixas de Texto	Média
2,0	0,02	73,46%	84,95%	83,75%	90,84%	83,25%
0,5	0,02	74,80%	86,31%	60,25%	96,07%	79,35%
0,5	0,40	38,52%	89,55%	26,25%	82,06%	59,09%
2,0	0,05	65,98%	80,00%	89,75%	77,57%	78,32%
1,0	0,30	65,98%	85,23%	74,75%	97,38%	80,83%

Como pode-se observar na coluna 7 da Tabela 4.9, o melhor do desempenho geral foi obtido para  $\alpha = 2,0$  e  $q_1 = 0,02$  resultando em um desempenho de 83,25%. Na coluna 3, verifica-se que o melhor desempenho para classificação de botões foi obtido para os valores  $\alpha = 0,5$  e  $q_1 = 0,02$ , possibilitando a classificação correta de 74,80% botões. A coluna 4 nos mostra que o melhor desempenho para classificação de ícones foi obtido para  $\alpha = 0,5$  e  $q_1 = 0,4$ , resultando em 89,55% de desempenho. O melhor resultado para agrupar áreas de texto foi obtida com valores  $\alpha = 2,0$  e  $q_1 = 0,05$ , como pode-se observar na coluna 5 da Tabela 4.9. Finalmente, observando a coluna 6, encontra-se o melhor desempenho do algoritmo para caixas de texto com os valores  $\alpha = 1,0$  e  $q_1 = 0,30$ .

#### 4.4.6 Classificação com Distância de Kullback-Leiber Automatizada

Neste último experimento, a classificação é feita utilizando-se a forma estendida da  $q$ -distância de Kullback-Leiber proposta por Rodrigues e Giraldi conforme a Equação (3.2) com o cálculo de  $q_1$  feito de forma automatizada. O resultado obtido com esta técnica depende dos valores de  $\alpha$  utilizados. Então, este algoritmo foi executado para diferentes valores de  $\alpha$ , variando entre 0,5 e 9. No gráfico da Figura 4.21, o eixo Y representa o desempenho do método para cada uma das classes estudadas. Os detalhes dos resultados do experimento estão listados na Tabela 4.10, onde em cada linha estão os desempenhos obtidos com a classificação dos objetos rotulados em cada classe.

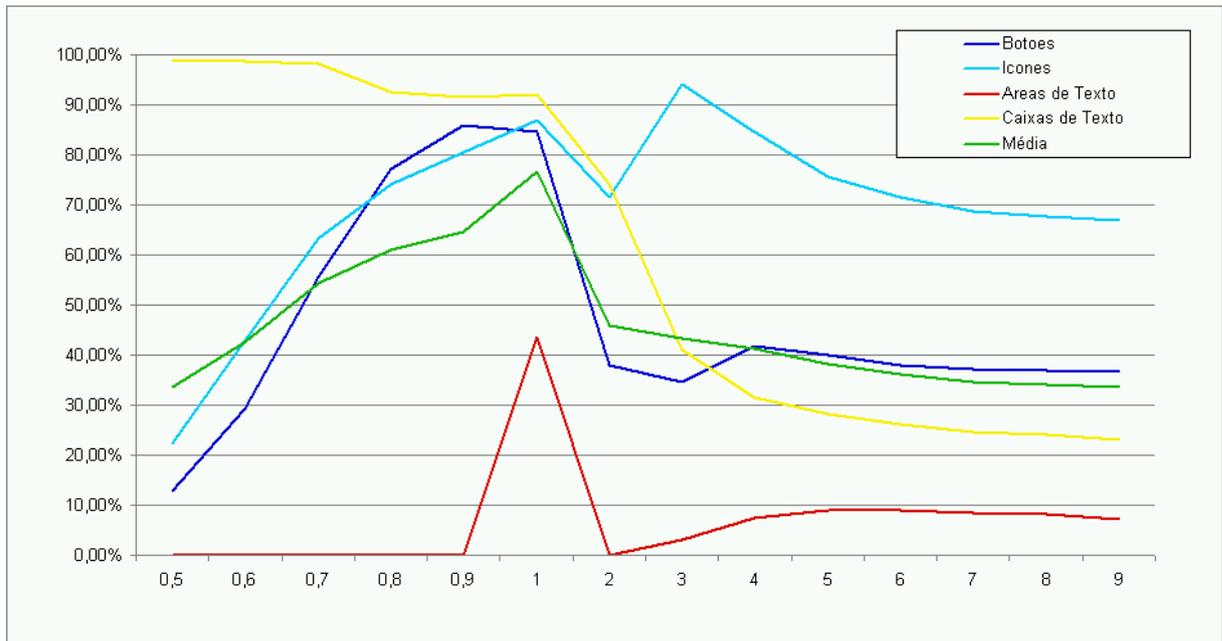


Figura 4.21: Resultados da classificação de objetos utilizando distância de Kullback-Leiber automatizada como métrica de similaridade

$\alpha$	Des. Bot.	Des. Ico.	Des. ATx.	Des. Cxs.
0,5	12,91%	22,25%	0,00 %	99,07%
0,6	29,30%	42,70%	0,00 %	98,69%
0,7	55,64%	63,42%	0,00 %	98,32%
0,8	77,25%	74,05%	0,00 %	92,52%
0,9	85,96%	80,54%	0,00 %	91,59%
1,0	84,53%	87,03%	43,50%	91,96%
2,0	38,01%	71,44%	0,00 %	74,02%
3,0	34,73%	94,23%	03,00%	40,93%
4,0	41,91%	84,50%	07,50%	31,59%
5,0	39,96%	75,59%	09,00%	28,22%
6,0	37,91%	71,53%	09,00%	26,17%
7,0	37,09%	68,65%	08,50%	24,49%
8,0	36,89%	67,57%	08,25%	24,11%
9,0	36,58%	66,85%	07,25%	23,18%

Tabela 4.10: Resultados da classificação de objetos utilizando distância de Kullback-Leiber automatizada como métrica de similaridade

A partir dos resultados exibidos na Tabela 4.10, pode-se observar que o melhor desempenho geral foi obtido com o parâmetro  $\alpha = 1,00$ , que proporcionou também a melhor classificação para as áreas de texto. No entanto, como pode-se observar na coluna 2 da Tabela 4.10, a melhor classificação para os botões é dada para  $\alpha = 0,9$ . Na coluna 3 da Tabela 4.10, verifica-se que para  $\alpha = 3,0$  foi obtido o melhor desempenho para ícones. Finalmente, observa-se na coluna mais à direita que a melhor classificação para caixas de texto com este algoritmo foi obtida com o parâmetro  $\alpha = 0,5$ .

#### 4.4.7 Comparação dos Algoritmos de Classificação

Após a execução dos experimentos com os métodos de classificação dos objetos, é possível fazer a comparação dos resultados obtidos. Os desempenhos dos algoritmos de classificação podem ser vistos no gráfico da Figura 4.22, onde o eixo Y representa o desempenho de cada método de classificação de objetos sintéticos utilizado. Na Tabela 4.11, estão listados os desempenhos obtidos para cada algoritmo e para cada classe estudada.

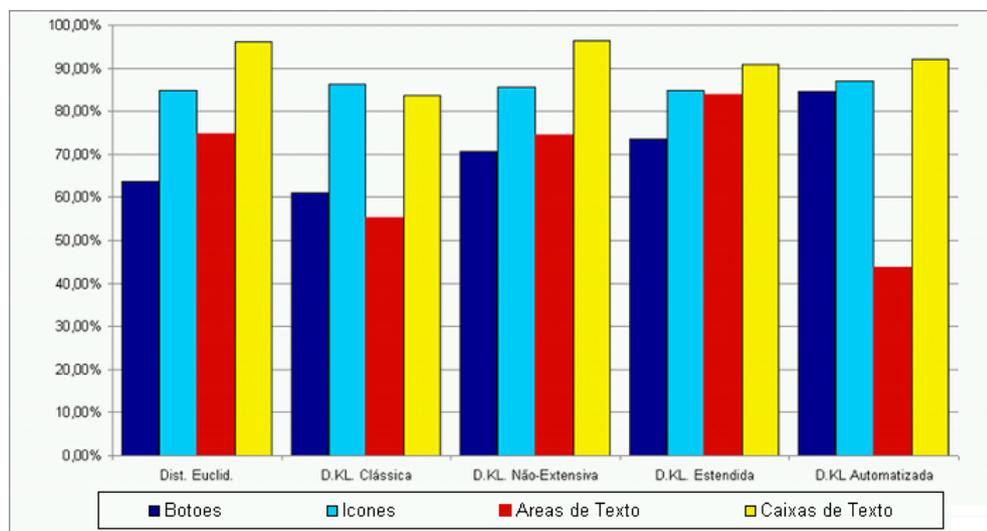


Figura 4.22: Comparação dos desempenhos dos algoritmos utilizados para classificação de objetos sintéticos

Tabela 4.11: Detalhe da comparação dos desempenhos dos algoritmos utilizados para classificação de objetos sintéticos

Algoritmo	Bot.	Ico.	ATx.	Cxs.	Obs.
Dist. Euclidiana.	63,62%	84,95%	74,50%	96,26%	
Dist. de KL. Clássica	61,16%	86,39%	55,00%	83,73%	
Dist. de KL. Não-Extensiva	70,59%	85,68%	74,25%	96,45%	$q = 0, 1$
Dist. de KL. Estendida	73,46%	84,95%	83,75%	90,84%	$\alpha=2$ e $q_1=0,02$
Dist. de KL. Automatizada	84,52%	87,02%	43,50%	91,96%	$\alpha=1$

A partir dos resultados obtidos, nota-se que, nas condições estudadas, a classificação de objetos utilizando a q-distancia estendida de Kullback-Leiber como métrica de similaridade oferece melhor desempenho geral em relação aos outros métodos estudados. No entanto, a classificação com a distância automatizada de Kullback-Leiber teve o melhor desempenho em três das quatro classes analisadas, obtendo resultado inferior somente para a detecção de áreas de texto, classe que foi melhor classificada com a q-distancia estendida de Kullback-Leiber.

## Capítulo 5

# CONCLUSÕES FINAIS

Este trabalho tem como objetivo, o estudo e comparação de técnicas de detecção e classificação de objetos sintéticos de interfaces gráficas baseadas em algoritmos de entropia e entropia não-extensiva. Dessa forma, apenas as propriedades visuais dos objetos como tamanho, aspecto, textura, posição, cores foram analisadas. Assim, foi realizada uma pesquisa sobre as principais técnicas de visão computacional e de como a entropia, em suas formas clássica e não-extensiva, poderia ser utilizada em conjunto com essas técnicas na detecção e na classificação de objetos sintéticos.

O método proposto para a detecção de objetos utiliza o conceito de entropia para segmentar a imagem em regiões proposto por Albuquerque *et al* (Albuquerque et al., 2004). Após a segmentação, as regiões são filtradas por um algoritmo de seleção de contornos que define os limites do objeto detectado. Para a classificação, a entropia relativa, ou distância de Kullback-Leiber, é utilizada como métrica de similaridade entre os vetores de características extraídas dos objetos de interesse e dos centróides de cada classe, onde a classificação do objeto é definida pela menor distância entre os vetores.

Os experimentos executados neste trabalho mostram que, para os objetos sintéticos em interfaces gráficas estudados, a detecção de objetos utilizando entropia, entropia não-extensiva e q-entropia estendida como algoritmo de segmentação teve desempenho inferior em relação à detecção de objetos utilizando limiar adaptativo como algoritmo segmentador, apesar da segmentação com entropia não-extensiva estendida ter um desempenho bastante satisfatório com imagens naturais (Rodrigues e Giraldi, 2009a). Este fato sugere que, entre as distribuições de cores do *foreground* e do *background*, não ocorrem relações de longo alcance temporal ou espacial, motivo que pode justificar os resultados não tão satisfatórios obtidos com uso da entropia não-extensiva como

método segmentador para as imagens sintéticas estudadas.

Foi também analisado o comportamento do método de classificação de objetos utilizando as formas clássica, não-extensiva, estendida e automatizada de entropia relativa como medida de similaridade. A classificação de objetos utilizando entropia não-extensiva teve o melhor desempenho em relação aos métodos de classificação. Com o valor de  $q$  fixado em 0,1, foi obtido o desempenho médio de quase 82%, que também foi o melhor resultado obtido na classificação com distância de Kullback-Leiber não-extensiva estendida quando os parâmetros entrópicos  $q_1$  e  $q_2$  tinham respectivamente os valores iguais a 0,02 e 0,04. Isso pode acontecer devido ao fato de que esta técnica faz uso de dois parâmetros de ajuste,  $q_1$  e  $\alpha$ , que permitem maior flexibilidade para atingir o melhor resultado. Dos métodos estudados, a forma estendida da distância de Kullback-Leiber é a mais indicada como métrica de similaridade para classificação de objetos sintéticos.

Finalmente, a combinação das técnicas estudadas, com exceção daquelas nas quais o parâmetro entrópico  $q$  é calculado automaticamente, pode ser implementada em um sistema para detecção e classificação de objetos artificiais em interfaces gráficas em tempo linear, uma vez que os algoritmos utilizados são executados em ordem  $O(n^2)$  para segmentação e  $O(n)$  para classificação.

## Capítulo 6

# TRABALHOS FUTUROS

Os estudos realizados neste trabalho podem ser continuados em outras pesquisas, dentre elas, o uso das técnicas apresentadas aqui para a implantação de sistemas de automação de testes em interfaces gráficas; sistemas de melhoria de acessibilidade das funcionalidades de um computador para pessoas com certo grau de deficiência, sistemas de reconhecimento de evidências em câmeras, sistemas de teste em celulares, entre outros.

Os algoritmos de detecção de objetos e classificação podem ser utilizados como *engines* do *software* de automação que foi simulado neste trabalho. Este software teria um certo grau de robustez em relação a deslocamento e a alteração de algumas das propriedades que os objetos podem sofrer. O fato de o automatizador fazer uso apenas de técnicas de visão computacional, tornará possível sua utilização em ambientes remotos, via qualquer programa de acesso remoto, como por exemplo VNC, Citrix® ou TerminalServices®.

Atualmente existe a tendência de migração de muitos aplicativos para a plataforma de celulares. Essa necessidade de migração também gera a necessidade de execução de testes para garantir que as aplicações migradas vão se comportar como previsto na nova plataforma. Já existem programas de acesso remoto como VNC para celulares, assim pode-se fazer total uso do automatizador proposto no parágrafo anterior para automação dos testes nesta nova plataforma.

Quando são realizados testes em dispositivos de Hardware que disponibilizam informações em telas, a captura da informação exibida na tela é essencial para verificação do resultado do teste. Durante a execução dos testes nesses dispositivos, a parada da execução para fazer captura de evidências nem sempre é viável, assim, pode-se, por meio de uma câmera e o algoritmo de detecção de objetos proposto, criar um programa de auxílio a testes manuais, onde é acionada uma rotina

de captura da imagem da câmera, toda vez que algum objeto gráfico pré-definido, como um ícone por exemplo, apareça no *display* do dispositivo.

Fazendo-se uso do método de detecção de objetos, é possível criar sistemas de melhoria de acessibilidade das funcionalidades de um computador à pessoas com algum grau de deficiência, de modo que os objetos gráficos exibidos na tela vão tendo suas dimensões aumentadas na medida que o ponteiro do mouse se aproxime do centro do objeto, tornando-o assim mais fácil de ser visualizado. Neste conceito é possível ainda utilizar avisos de voz informando quais tipos de objetos estão ao redor da coordenada do mouse, ajudando pessoas com alguma deficiência visual a utilizar sistemas operacionais gráficos.

# Referências Bibliográficas

- Ahmad, A.; Dey, L. A feature selection technique for classificatory analysis. *Pattern Recognition Letters*, 2004.
- Albuquerque, M.; Esquef, I.; Mello, A. Image thresholding using tsallis entropy. *Journal of Statistical Physics*, 25:1059–1065, 2004.
- Amant, R. Interface agents as surrogate users. *Intelligence*, 11(2):28–38, 2000.
- Amant, R.; Dudani, A. An environment for user interface softbots. In *AVI '02: Proceedings of the Working Conference on Advanced Visual Interfaces*, pages 119–122, 2002.
- Amant, R.; Riedl, M. A perception/action substrate for cognitive modeling in hci. *Int. J. Hum.-Comput. Stud.*, 55(1):15–39, 2001.
- Amant, R.; Zettlemoyer, L. The user interface as an agent environment. In *AGENTS '00: Proceedings of the fourth international conference on Autonomous agents*, pages 483–490, 2000.
- Ballard, D.; Brown, C. *Computer Vision*. Prentice Hall, 1982.
- Bastos, L.; Liatsis, P.; Conci, A. Automatic texture segmentation based on k-means clustering and efficient calculation of co-occurrence features. In *Systems, Signals and Image Processing, 2008. IWSSIP 2008. 15th International Conference*, pages 141–144, 2008.
- Beck, C. Non-extensive estatistical mechanics approach to fully develop hydrodynamic turbulence. *Chaos, Solutions and Fractals*, 13:499–506, 2002.
- Bernardo, P.; Kon, F. A importância dos testes automatizados. *Engenharia de Software Magazine*, 3:54–57, 2008.
- Bishop, C. *Neural Networks for Pattern Recognition*. Oxford, 1995.
- Blayvas, I.; Bruckstein, A.; Kimmel, R. Efficient computation of adaptive threshold surfaces for image binarization. *Pattern Recognition*, 39(1):89–101, 2006.

- Boltzmann, L. Further studies on the thermal equilibrium of gas molecules. *Wissenschaftliche Abhandlungen*, 1:275–370, 1872.
- Bolzan, M.; Sá, L.; Ramos, F.; Neto, C.; Rosa, R. Modelo da entropia generalizada aplicada aos sinais turbulentos medidos na camada limite superficial do pantanal. Trabalho do DCM - LAC / INPE, 2004.
- Cao, K. L.; Bonnet, A.; Gadat, S. Multiclass classification and gene selection with a stochastic algorithm. *Computational Statistics and Data Analysis*, 2009.
- Chen, J.; Wang, C.; Wang, R. Adaptive binary tree for fast svm multiclass classification. *Neuro-computing*, 2009.
- Clausius, R. *The Mechanical Theory of Heat, (2nd ed)*. London: Macmillan And Co., 1879.
- Coppi, R.; D’Urso, P. Fuzzy unsupervised classification of multivariate time trajectories with the shannon entropy regularization. *Comput. Stat. Data Anal.*, 50(6):1452–1477, 2006.
- Esquef, I. Tecnicas de entropia em processamento de imagens. Tese de mestrado, Centro Brasileiro de Pesquisas Fisicas, 2002.
- Forsyth, D.; Ponce, J. *Computer Vision: A Modern Approach*. Prentice Hall, 2002.
- Gentile, C.; Camps, O.; Sznaier, M. Segmentation for robust tracking in the presence of severe occlusion. *Image Processing, IEEE Transactions*, 13:166–178, 2004.
- Gibbs, J. A method of geometrical representation of the thermodynamic properties of substances by means of surfaces. *Transactions of the Connecticut Academy*, 2:382–404, 1873.
- Giraldi, G.; Rodrigues, P.; Marturelli, L. S.; Silva, R. L. S. *Improving the Initialization, Convergence and Memory Utilization for Deformable Models*, volume I: Segmentation Models, Part A of *Hand Book of Image Analysis*, chapter Chapter 7, pages 359–414. Kluwer Academic/Plenum, New York, Boston, Dordrecht, London, Moscow, 1nd edition, 2005.
- Gonzalez, R.; Woods, R. *Digital Image Processing*. Prentice Hall, 2001.
- Gouttière, C.; Lemaur, G.; Coninck, J. De. Influence of filter sphericity on the detection of singularities in synthetic images. *Signal Processing*, 87(3):552 – 561, 2007.
- Jain, A. K. Data clustering: 50 years beyond k-means. Technical Report MSU-CSE-09-11, Department of Computer Science, Michigan State University, East Lansing, Michigan, March 2009.

- Jensen, F.; Nielsen, T. *Bayesian Networks and Decision Graphs*. Springer, 2007.
- Karmeshu; Krishnamachari, A. *Neural Information Processing*. Springer Berlin / Heidelberg, 2004.
- Kass, M.; Witkin, A. P.; Terzopoulos, D. Snakes: Active contour models. *International Journal of Computer Vision*, pages 321–331, 1988.
- Kullback, S.; Leibler, R. On information and sufficiency. *The Annals of Mathematics and Statistics*, vol. 22(1):79–86, March 1951.
- Macqueen, J. B. Some methods of classification and analysis of multivariate observations. In *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*, pages 281–297, 1967.
- Martin, S.; Morison, G.; Nailon, W.; Durrani, T. Fast and accurate image registration using tsallis entropy and simultaneous perturbation stochastic approximation. *ELECTRONICS LETTERS*, 40(10), May 2004.
- Martínez-Finkelshtein, A.; Sánchez-Lara, J. Shannon entropy of symmetric pollaczek polynomials. *J. Approx. Theory*, 145(1):55–80, 2007.
- Nakamichi, A.; Morikawa, M. Is galaxy distribution non-extensive and non-gaussian? *Physica A: Statistical Mechanics and its Applications*, 341:215 – 233, 2004.
- Niemisto, A.; Korpelainen, T.; Saleem, R.; Yli-Harja, O.; Aitchison, J.; Shmulevich, I. A k-means segmentation method for finding 2-d object areas based on 3-d image stacks obtained by confocal microscopy. In *Engineering in Medicine and Biology Society, 2007. EMBS 2007. 29th Annual International Conference of the IEEE*, pages 5559–5562, 2007.
- Noyer, J. C.; Lanvin, P.; Benjelloun, M. Non-linear matched filtering for object detection and tracking. *Pattern Recognition Letters*, 25(6):655 – 668, 2004.
- Ouyang, J.; Patel, N.; Sethi, I. Induction of multiclass multifeature split decision trees from distributed data. *Pattern Recognition*, 2009.
- Pandit, M.; Kalbag, S. The selection recognition agent: instant access to relevant information and operations. In *IUI '97: Proceedings of the 2nd international conference on Intelligent user interfaces*, pages 47–52, New York, NY, USA, 1997. ACM.

- Pereira, V. Autolocalização qualitativa para um robô móvel utilizando informação de sombras. Tese de mestrado, Centro Universitário da FEI, 2009.
- Pham, N.; Morrison, A.; Schwock, J.; Aviel-Ronen, S.; Iakovlev, V.; Tsao, M.; Ho, J.; Hedley, D. Quantitative image analysis of immunohistochemical stains using a cmyk color model. *Diagnostic Pathology*, 2, February 2007.
- Pi, M.; Zhang, H. Two-stage image segmentation by adaptive thresholding and gradient watershed. In *CRV '05: Proceedings of the 2nd Canadian conference on Computer and Robot Vision*, pages 57–64. IEEE Computer Society, 2005.
- Piera, F. J.; Parada, P. On convergence properties of shannon entropy. *Probl. Inf. Transm.*, 45(2): 75–94, 2009.
- Rodrigues, P. *Um Modelo Bayesiano Combinando Análise Semântica Latente e Atributos Espaciais para Recuperação de Informação Visual*. Tese de doutorado, Universidade Federal de Minas Gerais, 2003.
- Rodrigues, P.; Giraldi, G. Improving the non-extensive tsallis non-extensive medical image segmentation based on tsallis entropy. *Pattern Analysis and Application*, Submitted, 2009a.
- Rodrigues, P.; Giraldi, G. Computing the q-index for tsallis non-extensive image segmentation. In SBC, editor, *XXIInd Brazilian Symposium on Computer Graphics and Image Processing (Sibgrapi 2009)*, volume To Appear, 2009b.
- Rodrigues, P.; Giraldi, G.; Chang, R.; Suri, J. Object tracking in image sequence combining hausdorff distance, non-extensive entropy in level set formulation. *Topics in Biomedical Engineering International Deformable Models*, pages 477–515, 2005.
- Rodrigues, P.; Giraldi, G.; Chang, R.; Suri, J. Non-extensive entropy for cad systems of breast cancer images. In Society, IEEE Computer, editor, *Proceedings of Brazilian Simposium on Computer Graphics and Image Processing*, pages 121–128, 2006a.
- Rodrigues, P.; Giraldi, G.; Suri, J. *Object Tracking Combining Haudorff Distance, HSV Histogram and Non-Extensive Entropy*, volume 1, chapter 15. Springer, New York, 2006b.
- Rodrigues, P.; Giraldi, G.; Suri, J. *Object Tracking Combining Haudorff Distance, HSV Histogram and Non-Extensive Entropy*, volume 1, chapter 15. Springer, 2006c.

- Rodrigues, P.; Giraldi, G.; Suri, J.; Chang, R. Non-extensive entropy for cad systems for breast cancer images. In Society, IEEE Computer, editor, *International Symposium on Computer Graphics and Image processing, SIBIGRAPI'06*, pages 121–128, Manaus, 2006d.
- Rosenberger, C. Adaptive evaluation of image segmentation results. *Pattern Recognition, International Conference on*, 2:399–402, 2006.
- S. Gunal, R. Edizkan. Subspace based feature selection for pattern recognition. *Information Sciences*, 2008.
- Sang, N.; Li, H.; Peng, W.; Zhang, T. Knowledge-based adaptive thresholding segmentation of digital subtraction angiography images. *Image Vision Comput.*, 25(8):1263–1270, 2007.
- Santos, D. N. Descrição automática das diferenças estatisticamente mais relevantes entre neuroimagens de pacientes com esquizofrenia e controles saudáveis. Tese de mestrado, Centro Universitário da FEI, 2009.
- Sezgin, M.; Sankur, B. Survey over image thresholding techniques and quantitative performance evaluation. *Journal of Electronic Imaging*, 13(1):146–168, 2004.
- Shannon, C. A mathematical theory of communication. *The Bell System Technical Journal*, 27: 379–423, 623–656, 1948.
- Shannon, C.; Weaver, W. The mathematical theory of communication. *University of Illinois Press, Urbana*, 1948.
- Shapiro, L.; Stockman, G. *Computer Vision*. Prentice Hall, 2000.
- Sziranyi, T.; Zerubia, J.; Czuni, L.; Geldreich, D.; Kato, Z. Image segmentation using markov random field model in fully parallel cellular network architectures. *Real-Time Imaging*, 6, 2000.
- Tadaki, K. The tsallis entropy and the shannon entropy of a universal probability. *CoRR*, abs/0805.0154, 2008.
- Taruya, A.; Sakagami, M. Gravitational catastrophe and tsallis' generalized entropy of self-gravitating systems. *Physica A*, 307:185–206, 2002.
- Tavares, A. Aspectos matemáticos da entropia. Tese de mestrado, Universidade de Aveiro, 2003.
- Tissainayagama, P.; Suterb, D. Object tracking in image sequences using point features. *Pattern Recognition*, 38:105–113, 2005.

- Tong, S.; Malhotra, A.; Zhu, Y.; Thakor, N. Parameterized entropy analysis of eeg following hypoxic-ischemic brain injury. *Physics Letters A*, 314(5-6):354 – 361, 2003.
- Tsallis, C. Possible generalization of boltzmann-gibbs statistics. *Journal of Statistical Physics*, 52: 479–487, 1988.
- Tsallis, C. Non-extensive statistical mechanics and its applications. *Series Lecture Notes in Physics, Berlin*, 2001.
- Villegas, O.; Elias, R.; Villela, R.; Sanchez, C.; Salazar, A. Edging out the competition: Lossy image coding with wavelets and contourlets. *IEEE Potentials*, 27:39–44, 2008.
- Walt, C.; Barnard, E. Data characteristics that determine classifier performance. *SAIEE Africa Research Journal*, 98:87–93, 2007.
- Wang, X.; Leiser, M. K-means clustering for multispectral images using floating-point divide. In *FCCM '07: Proceedings of the 15th Annual IEEE Symposium on Field-Programmable Custom Computing Machines*, pages 151–162, Washington, DC, USA, 2007. IEEE Computer Society.
- Wu, J.; Xiong, H.; Chen, J. Adapting the right measures for k-means clustering. In *KDD '09: Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 877–886, New York, NY, USA, 2009. ACM.
- Zhang, Y.; Wu, L. Pattern recognition via pcnn and tsallis entropy. *Sensors*, 8(11):7518–7529, 2008.
- Zhu, Y.; Yu, J.; Jia, C. Initializing k-means clustering using affinity propagation. In *HIS '09: Proceedings of the 2009 Ninth International Conference on Hybrid Intelligent Systems*, pages 338–343, Washington, DC, USA, 2009. IEEE Computer Society.

API	— <i>Application Programming Interface</i> é um conjunto de funções implementadas em um software para permitir que outros programas acessem suas funções.
Binarizacao	— Ato de separar uma imagem em duas classes.
Citrix <sup>®</sup>	— Programa de acesso remoto que permite utilizar a interface gráfica de um computador remotamente.
<i>Degrade</i>	— Variação suave entre duas cores ou tons.
Hardware	— do dicionário Houaiss - "... conjunto dos componentes físicos (material eletrônico, placas, monitor, equipamentos periféricos etc) de um computador..."
Heurística	— do dicionário Houaiss - "...método de investigação baseado na aproximação progressiva de um dado problema..."
Inteligência artificial	— do dicionário Houaiss - "... ramo da informática que visa dotar os computadores da capacidade de simular certos aspectos da inteligência humana, tais como aprender com a experiência, inferir a partir de dados incompletos, tomar decisões em condições de incerteza e compreender a linguagem falada, entre outros."
<i>Linux</i>	— Sistema Operacional de código aberto baseado em UNIX.
<i>OpenCV</i>	— Biblioteca com funções de visão computacional fornecida pela <i>Intel</i> .
<i>Pixel</i>	— Unidade básica de amostra de uma imagem.
Segmentação	— Ato de separar por característica.
Sistema Operacional	— Conjunto de programas que fazem a abstração do hardware tornando mais simples aos programas fazer uso dos dispositivos físicos dos computadores.
Software	— do dicionário Houaiss - "...conjunto de componentes lógicos de um computador ou sistema de processamento de dados; programa, rotina ou conjunto de instruções que controlam o funcionamento de um computador; suporte lógico..."
Terminal Services <sup>®</sup>	— Programa de acesso remoto que permite utilizar a interface gráfica de um computador remotamente disponível com o sistema operacional Windows <sup>®</sup> .
UNIX	— Sistema operacional criado pela empresa AT&T <sup>®</sup> .
VNC	— Protocolo aberto de comunicação que permite acesso à interface gráfica de um computador remotamente disponível.
Windows <sup>®</sup>	— Sistema operacional gráfico proprietário da Microsoft <sup>®</sup> .

# Apêndice A

## SEGMENTAÇÃO COM Q-ENTROPIA ESTENDIDA

Resultados obtidos na condução dos experimentos realizados com a segmentação por meio de q-entropia estendida proposta por Rodrigues e Giraldi (Rodrigues e Giraldi, 2009a).

Tabela A.1: Resultados da detecção de objetos utilizando q-entropia estendida como algoritmo de segmentação

$\alpha$	$q_1$	Botões	Ícones	Áreas de Texto	Caixas de Texto
0,5	0,1	50,00%	57,14%	33,33%	70,00%
0,5	0,2	43,79%	51,90%	45,83%	67,78%
0,5	0,3	40,69%	50,48%	45,83%	66,67%
0,5	0,4	38,97%	51,43%	45,83%	64,44%
0,5	0,5	38,62%	48,10%	45,83%	64,44%
0,5	0,6	38,62%	43,81%	45,83%	62,22%
0,5	0,7	38,62%	37,14%	41,67%	63,33%
0,5	0,8	34,48%	31,90%	41,67%	53,33%
0,5	0,9	33,79%	30,48%	41,67%	47,78%
0,5	1,0	50,69%	52,86%	58,33%	74,44%
0,5	1,1	27,59%	20,00%	37,50%	36,67%
0,5	1,2	27,59%	11,90%	29,17%	32,22%
0,5	1,3	24,48%	11,43%	29,17%	26,67%
0,5	1,4	23,79%	10,00%	25,00%	28,89%
0,5	1,5	21,03%	11,43%	16,67%	30,00%
0,5	1,6	19,31%	07,14%	12,50%	25,56%
0,5	1,7	17,59%	10,95%	08,33%	21,11%
0,5	1,8	14,83%	12,38%	12,50%	16,67%
0,5	1,9	15,52%	13,33%	12,50%	16,67%

Tabela A.1 – continuação da página anterior

$\alpha$	$q_1$	Botões	Ícones	Áreas de Texto	Caixas de Texto
0,5	2,0	00,00%	00,00%	00,00%	00,00%
0,6	0,1	50,00%	57,14%	33,33%	70,00%
0,6	0,2	46,21%	54,29%	45,83%	67,78%
0,6	0,3	41,38%	50,95%	45,83%	65,56%
0,6	0,4	39,66%	48,10%	45,83%	64,44%
0,6	0,5	39,31%	48,10%	45,83%	63,33%
0,6	0,6	39,31%	45,71%	45,83%	61,11%
0,6	0,7	40,00%	38,57%	45,83%	61,11%
0,6	0,8	38,62%	36,67%	45,83%	61,11%
0,6	0,9	36,21%	32,38%	41,67%	54,44%
0,6	1,0	50,69%	52,86%	58,33%	74,44%
0,6	1,1	30,34%	28,10%	41,67%	41,11%
0,6	1,2	29,31%	25,24%	41,67%	45,56%
0,6	1,3	27,59%	19,52%	33,33%	46,67%
0,6	1,4	27,93%	19,52%	29,17%	37,78%
0,6	1,5	28,28%	18,57%	29,17%	37,78%
0,6	1,6	25,86%	20,48%	29,17%	33,33%
0,6	1,7	27,59%	17,62%	37,50%	25,56%
0,6	1,8	28,97%	18,57%	41,67%	31,11%
0,6	1,9	26,90%	18,57%	37,50%	30,00%
0,6	2,0	27,24%	19,05%	37,50%	30,00%
0,7	0,1	48,97%	57,14%	33,33%	68,89%
0,7	0,2	45,86%	54,29%	45,83%	72,22%
0,7	0,3	45,17%	50,95%	45,83%	66,67%
0,7	0,4	42,07%	47,14%	45,83%	66,67%
0,7	0,5	41,03%	48,57%	45,83%	65,56%
0,7	0,6	43,79%	48,10%	45,83%	65,56%
0,7	0,7	43,10%	45,24%	45,83%	58,89%
0,7	0,8	43,10%	40,95%	45,83%	60,00%
0,7	0,9	41,03%	38,57%	45,83%	66,67%
0,7	1,0	50,69%	52,86%	58,33%	74,44%
0,7	1,1	37,59%	31,43%	41,67%	50,00%
0,7	1,2	34,14%	32,38%	41,67%	46,67%
0,7	1,3	33,10%	28,10%	41,67%	42,22%
0,7	1,4	30,34%	27,62%	37,50%	42,22%
0,7	1,5	30,34%	20,95%	41,67%	38,89%
0,7	1,6	30,69%	21,43%	41,67%	40,00%

Tabela A.1 – continuação da página anterior

$\alpha$	$q_1$	Botões	Ícones	Áreas de Texto	Caixas de Texto
0,7	1,7	30,69%	21,90%	41,67%	40,00%
0,7	1,8	30,34%	22,38%	41,67%	41,11%
0,7	1,9	30,34%	23,33%	41,67%	40,00%
0,7	2,0	30,00%	20,95%	45,83%	38,89%
0,8	0,1	49,31%	57,62%	33,33%	74,44%
0,8	0,2	48,28%	57,14%	45,83%	71,11%
0,8	0,3	44,48%	47,62%	45,83%	65,56%
0,8	0,4	46,55%	49,05%	45,83%	67,78%
0,8	0,5	48,28%	50,95%	45,83%	66,67%
0,8	0,6	47,93%	50,48%	45,83%	66,67%
0,8	0,7	46,90%	46,67%	50,00%	64,44%
0,8	0,8	46,90%	44,76%	50,00%	64,44%
0,8	0,9	44,83%	41,43%	45,83%	60,00%
0,8	1,0	50,69%	52,86%	58,33%	74,44%
0,8	1,1	44,14%	40,48%	45,83%	63,33%
0,8	1,2	43,79%	38,10%	45,83%	60,00%
0,8	1,3	41,03%	35,24%	41,67%	51,11%
0,8	1,4	39,66%	30,00%	41,67%	50,00%
0,8	1,5	38,28%	31,90%	41,67%	50,00%
0,8	1,6	37,24%	27,62%	41,67%	50,00%
0,8	1,7	34,83%	25,71%	41,67%	47,78%
0,8	1,8	34,83%	23,81%	41,67%	47,78%
0,8	1,9	34,83%	23,33%	45,83%	45,56%
0,8	2,0	35,86%	24,29%	45,83%	46,67%
0,9	0,1	48,62%	57,62%	33,33%	75,56%
0,9	0,2	48,28%	56,67%	45,83%	68,89%
0,9	0,3	47,93%	50,95%	45,83%	63,33%
0,9	0,4	47,24%	49,05%	45,83%	66,67%
0,9	0,5	50,69%	48,10%	50,00%	66,67%
0,9	0,6	51,72%	49,05%	50,00%	67,78%
0,9	0,7	49,66%	47,14%	45,83%	65,56%
0,9	0,8	50,00%	45,71%	45,83%	64,44%
0,9	0,9	48,28%	44,76%	45,83%	67,78%
0,9	1,0	50,69%	52,86%	58,33%	74,44%
0,9	1,1	47,59%	42,86%	50,00%	66,67%
0,9	1,2	47,93%	41,43%	50,00%	62,22%
0,9	1,3	46,55%	39,05%	50,00%	58,89%

Tabela A.1 – continuação da página anterior

$\alpha$	$q_1$	Botões	Ícones	Áreas de Texto	Caixas de Texto
0,9	1,4	46,55%	36,67%	50,00%	61,11%
0,9	1,5	46,55%	36,19%	50,00%	52,22%
0,9	1,6	43,79%	31,90%	45,83%	52,22%
0,9	1,7	42,07%	30,00%	50,00%	48,89%
0,9	1,8	41,38%	28,57%	45,83%	47,78%
0,9	1,9	42,41%	27,62%	45,83%	48,89%
0,9	2,0	40,34%	28,10%	45,83%	54,44%
1,0	0,1	48,28%	58,10%	50,00%	77,78%
1,0	0,2	47,24%	57,62%	41,67%	72,22%
1,0	0,3	45,86%	51,90%	37,50%	70,00%
1,0	0,4	50,34%	50,48%	37,50%	68,89%
1,0	0,5	50,00%	47,62%	37,50%	68,89%
1,0	0,6	51,03%	48,10%	41,67%	65,56%
1,0	0,7	50,00%	48,57%	58,33%	65,56%
1,0	0,8	48,62%	48,57%	62,50%	72,22%
1,0	0,9	50,34%	48,57%	58,33%	70,00%
1,0	1,0	50,34%	54,29%	58,33%	67,78%
1,0	1,1	51,03%	47,14%	62,50%	67,78%
1,0	1,2	51,38%	43,33%	62,50%	68,89%
1,0	1,3	50,00%	40,48%	54,17%	67,78%
1,0	1,4	51,03%	39,05%	50,00%	66,67%
1,0	1,5	51,72%	38,57%	50,00%	68,89%
1,0	1,6	50,69%	38,57%	54,17%	67,78%
1,0	1,7	50,34%	39,05%	54,17%	66,67%
1,0	1,8	50,00%	38,57%	54,17%	67,78%
1,0	1,9	48,97%	37,14%	50,00%	67,78%
1,0	2,0	48,62%	36,67%	50,00%	67,78%
2,0	0,1	51,72%	56,67%	37,50%	62,22%
2,0	0,2	49,66%	44,76%	41,67%	70,00%
2,0	0,3	52,41%	42,86%	54,17%	77,78%
2,0	0,4	51,03%	41,90%	62,50%	70,00%
2,0	0,5	11,38%	09,52%	25,00%	12,22%
2,0	0,6	43,10%	48,10%	66,67%	73,33%
2,0	0,7	39,31%	42,86%	58,33%	77,78%
2,0	0,8	39,66%	40,00%	58,33%	74,44%
2,0	0,9	40,00%	37,14%	45,83%	73,33%
2,0	1,0	50,69%	52,86%	58,33%	74,44%

Tabela A.1 – continuação da página anterior

$\alpha$	$q_1$	Botões	Ícones	Áreas de Texto	Caixas de Texto
2,0	1,1	38,28%	34,76%	41,67%	72,22%
2,0	1,2	38,97%	33,33%	41,67%	73,33%
2,0	1,3	38,28%	32,38%	41,67%	74,44%
2,0	1,4	37,24%	32,86%	45,83%	75,56%
2,0	1,5	35,86%	30,48%	41,67%	76,67%
2,0	1,6	35,86%	30,95%	41,67%	76,67%
2,0	1,7	34,14%	30,95%	41,67%	76,67%
2,0	1,8	34,48%	30,00%	50,00%	76,67%
2,0	1,9	34,48%	30,00%	50,00%	76,67%
2,0	2,0	34,14%	30,00%	50,00%	76,67%
3,0	0,1	51,38%	50,48%	37,50%	68,89%
3,0	0,2	49,66%	44,76%	62,50%	72,22%
3,0	0,3	45,17%	41,90%	70,83%	72,22%
3,0	0,4	44,83%	40,95%	75,00%	74,44%
3,0	0,5	38,97%	43,33%	58,33%	80,00%
3,0	0,6	36,90%	41,43%	45,83%	75,56%
3,0	0,7	38,28%	36,19%	41,67%	78,89%
3,0	0,8	36,21%	32,86%	41,67%	76,67%
3,0	0,9	36,21%	33,33%	45,83%	77,78%
3,0	1,0	50,69%	52,86%	58,33%	74,44%
3,0	1,1	34,48%	30,48%	41,67%	81,11%
3,0	1,2	34,48%	29,52%	41,67%	81,11%
3,0	1,3	32,76%	29,05%	50,00%	81,11%
3,0	1,4	33,10%	29,05%	50,00%	81,11%
3,0	1,5	33,45%	29,05%	50,00%	81,11%
3,0	1,6	31,03%	29,05%	50,00%	81,11%
3,0	1,7	30,69%	29,05%	50,00%	80,00%
3,0	1,8	30,69%	29,05%	50,00%	80,00%
3,0	1,9	30,69%	28,57%	50,00%	80,00%
3,0	2,0	31,03%	28,57%	50,00%	80,00%
4,0	0,1	50,69%	43,33%	50,00%	74,44%
4,0	0,2	44,83%	41,90%	70,83%	73,33%
4,0	0,3	44,48%	38,57%	75,00%	77,78%
4,0	0,4	37,93%	45,71%	70,83%	84,44%
4,0	0,5	38,62%	36,19%	50,00%	78,89%
4,0	0,6	36,21%	34,29%	45,83%	80,00%
4,0	0,7	36,21%	32,86%	45,83%	77,78%

Tabela A.1 – continuação da página anterior

$\alpha$	$q_1$	Botões	Ícones	Áreas de Texto	Caixas de Texto
4,0	0,8	34,83%	31,43%	41,67%	81,11%
4,0	0,9	34,83%	30,48%	50,00%	81,11%
4,0	1,0	50,69%	52,86%	58,33%	74,44%
4,0	1,1	33,10%	29,52%	50,00%	81,11%
4,0	1,2	33,45%	29,05%	50,00%	81,11%
4,0	1,3	31,03%	29,05%	50,00%	81,11%
4,0	1,4	30,69%	28,57%	50,00%	80,00%
4,0	1,5	30,69%	28,57%	50,00%	80,00%
4,0	1,6	31,03%	28,57%	50,00%	80,00%
4,0	1,7	31,03%	28,57%	50,00%	80,00%
4,0	1,8	31,38%	28,10%	50,00%	80,00%
4,0	1,9	31,72%	28,10%	50,00%	80,00%
4,0	2,0	31,72%	28,10%	50,00%	80,00%
5,0	0,1	50,34%	47,62%	62,50%	73,33%
5,0	0,2	11,38%	09,52%	25,00%	12,22%
5,0	0,3	37,59%	42,38%	66,67%	82,22%
5,0	0,4	36,90%	35,24%	50,00%	78,89%
5,0	0,5	35,86%	32,38%	54,17%	81,11%
5,0	0,6	34,83%	30,48%	50,00%	81,11%
5,0	0,7	34,83%	30,00%	50,00%	81,11%
5,0	0,8	34,83%	30,00%	50,00%	81,11%
5,0	0,9	33,10%	29,52%	50,00%	81,11%
5,0	1,0	50,69%	52,86%	58,33%	74,44%
5,0	1,1	32,07%	29,05%	50,00%	80,00%
5,0	1,2	30,69%	28,10%	50,00%	80,00%
5,0	1,3	31,03%	28,57%	50,00%	80,00%
5,0	1,4	31,03%	28,57%	50,00%	80,00%
5,0	1,5	31,03%	28,57%	50,00%	80,00%
5,0	1,6	31,38%	28,10%	50,00%	80,00%
5,0	1,7	31,72%	28,10%	50,00%	80,00%
5,0	1,8	31,72%	28,10%	50,00%	80,00%
5,0	1,9	32,07%	28,10%	50,00%	80,00%
5,0	2,0	32,41%	28,10%	50,00%	80,00%

## Apêndice B

# CLASSIFICAÇÃO COM DISTÂNCIA ESTENDIDA DE KULLBACK-LEIBER

Resultados obtidos na condução dos experimentos realizados para a classificação por meio de distância não-extensiva de Kullback-Leiber estendida.

Tabela B.1: Resultados da classificação de objetos utilizando distância de Kullback-Leiber estendida como métrica de similaridade

$\alpha$	$q_1$	Botões	Ícones	Áreas de Texto	Caixas de Texto
0,5	0,01	74,69%	85,86%	65,75%	96,07%
0,6	0,01	74,49%	85,86%	66,25%	96,07%
0,7	0,01	74,18%	85,86%	67,25%	96,07%
0,8	0,01	74,18%	85,86%	70,75%	96,07%
0,9	0,01	74,18%	85,86%	72,00%	96,07%
1,0	0,01	74,08%	85,86%	72,50%	95,70%
2,0	0,01	73,36%	85,59%	78,75%	93,64%
3,0	0,01	73,77%	85,05%	83,50%	90,47%
4,0	0,01	72,54%	83,06%	87,00%	85,61%
5,0	0,01	70,29%	81,89%	88,25%	82,24%
6,0	0,01	67,93%	80,54%	89,50%	78,32%
7,0	0,01	65,16%	79,37%	88,00%	76,07%
8,0	0,01	62,60%	78,38%	87,50%	73,27%
9,0	0,01	61,78%	77,39%	85,25%	70,47%
0,5	0,02	74,80%	86,31%	60,25%	96,07%
0,6	0,02	74,49%	86,22%	62,25%	96,45%
0,7	0,02	74,18%	85,86%	64,50%	96,45%
0,8	0,02	73,77%	85,95%	66,75%	96,45%
0,9	0,02	73,57%	85,95%	70,50%	96,45%

Tabela B.1 – continuação da página anterior

$\alpha$	$q_1$	Botões	Ícones	Áreas de Texto	Caixas de Texto
1,0	0,02	73,46%	85,86%	73,00%	96,07%
2,0	0,02	73,46%	84,95%	83,75%	90,84%
3,0	0,02	69,57%	81,62%	88,25%	82,06%
4,0	0,02	64,45%	79,28%	88,75%	75,70%
5,0	0,02	61,07%	77,30%	85,00%	69,91%
6,0	0,02	60,86%	75,95%	80,00%	67,85%
7,0	0,02	59,73%	74,68%	76,00%	69,91%
8,0	0,02	58,40%	74,77%	72,00%	71,40%
9,0	0,02	56,97%	74,68%	69,25%	73,27%
0,5	0,05	74,59%	86,49%	54,50%	94,39%
0,6	0,05	73,77%	86,58%	56,00%	94,95%
0,7	0,05	72,75%	86,58%	57,75%	95,89%
0,8	0,05	72,85%	86,31%	60,75%	96,64%
0,9	0,05	72,54%	85,86%	66,00%	97,01%
1,0	0,05	71,72%	85,68%	73,50%	96,64%
2,0	0,05	65,98%	80,00%	89,75%	77,57%
3,0	0,05	59,73%	75,86%	79,75%	67,85%
4,0	0,05	56,05%	74,86%	69,75%	72,34%
5,0	0,05	51,13%	71,71%	58,75%	74,21%
6,0	0,05	40,98%	69,91%	50,00%	75,89%
7,0	0,05	32,48%	67,39%	40,25%	74,77%
8,0	0,05	22,13%	64,59%	38,75%	74,77%
9,0	0,05	17,52%	56,85%	37,00%	73,83%
0,5	0,08	74,28%	87,21%	51,00%	94,77%
0,6	0,08	74,18%	87,12%	51,50%	94,39%
0,7	0,08	72,95%	86,58%	54,50%	94,95%
0,8	0,08	72,75%	86,58%	57,75%	96,64%
0,9	0,08	71,41%	85,86%	62,25%	96,82%
1,0	0,08	71,11%	85,68%	74,25%	96,45%
2,0	0,08	59,73%	76,85%	83,50%	65,05%
3,0	0,08	53,18%	75,41%	68,50%	73,27%
4,0	0,08	41,50%	71,44%	49,75%	75,33%
5,0	0,08	25,92%	68,92%	37,50%	74,95%
6,0	0,08	15,27%	58,02%	33,50%	74,77%
7,0	0,08	06,05%	41,17%	35,75%	72,52%
8,0	0,08	03,79%	26,94%	37,75%	70,65%
9,0	0,08	02,66%	22,16%	40,50%	67,29%

Tabela B.1 – continuação da página anterior

$\alpha$	$q_1$	Botões	Ícones	Áreas de Texto	Caixas de Texto
0,5	0,10	74,49%	87,48%	48,50%	94,02%
0,6	0,10	73,77%	87,12%	51,00%	95,14%
0,7	0,10	72,75%	86,85%	52,00%	94,77%
0,8	0,10	71,82%	86,67%	56,75%	95,70%
0,9	0,10	71,11%	86,13%	60,25%	96,82%
1,0	0,10	70,59%	85,68%	74,25%	96,45%
2,0	0,10	58,09%	75,68%	78,00%	66,92%
3,0	0,10	48,05%	74,50%	57,00%	74,39%
4,0	0,10	28,28%	70,99%	35,00%	74,95%
5,0	0,10	13,22%	59,64%	31,75%	74,77%
6,0	0,10	04,30%	37,03%	34,00%	72,52%
7,0	0,10	02,87%	24,23%	38,50%	68,04%
8,0	0,10	01,13%	21,08%	38,25%	68,97%
9,0	0,10	01,54%	38,56%	35,50%	72,15%
0,5	0,20	71,62%	87,84%	38,50%	91,78%
0,6	0,20	72,75%	87,93%	39,25%	93,27%
0,7	0,20	72,95%	87,48%	45,00%	93,64%
0,8	0,20	71,52%	86,94%	51,00%	95,14%
0,9	0,20	69,88%	86,13%	55,25%	96,26%
1,0	0,20	68,85%	85,50%	74,00%	97,38%
2,0	0,20	43,03%	77,21%	50,75%	74,39%
3,0	0,20	05,94%	63,87%	26,75%	76,07%
4,0	0,20	01,33%	29,46%	35,00%	69,91%
5,0	0,20	22,03%	75,50%	51,75%	88,79%
6,0	0,20	54,00%	81,80%	18,00%	23,55%
7,0	0,20	47,44%	80,36%	14,25%	20,00%
8,0	0,20	41,09%	78,11%	13,50%	17,38%
9,0	0,20	35,86%	77,75%	12,50%	17,57%
0,5	0,30	61,78%	88,20%	32,25%	85,05%
0,6	0,30	64,55%	87,93%	34,25%	89,53%
0,7	0,30	68,85%	87,93%	39,75%	91,40%
0,8	0,30	71,21%	87,39%	41,50%	93,46%
0,9	0,30	69,16%	86,40%	51,25%	96,26%
1,0	0,30	65,98%	85,23%	74,75%	97,38%
2,0	0,30	09,43%	84,59%	25,75%	76,07%
3,0	0,30	02,77%	60,54%	30,25%	81,31%
4,0	0,30	56,35%	83,42%	20,75%	25,79%

Tabela B.1 – continuação da página anterior

$\alpha$	$q_1$	Botões	Ícones	Áreas de Texto	Caixas de Texto
5,0	0,30	47,13%	80,81%	14,25%	19,81%
6,0	0,30	37,40%	80,81%	14,25%	17,57%
7,0	0,30	32,79%	80,72%	12,25%	17,01%
8,0	0,30	29,41%	79,64%	11,50%	16,45%
9,0	0,30	27,66%	78,92%	11,25%	16,07%
0,5	0,40	38,52%	89,55%	26,25%	82,06%
0,6	0,40	52,46%	89,28%	28,50%	83,93%
0,7	0,40	62,40%	87,84%	33,00%	86,17%
0,8	0,40	68,14%	87,93%	39,50%	91,59%
0,9	0,40	68,55%	86,85%	46,00%	95,14%
1,0	0,40	64,75%	85,05%	74,25%	97,38%
2,0	0,40	04,00%	65,86%	26,50%	78,50%
3,0	0,40	58,61%	83,87%	25,25%	35,51%
4,0	0,40	45,70%	81,08%	14,00%	20,00%
5,0	0,40	35,45%	81,08%	13,50%	17,38%
6,0	0,40	30,64%	80,09%	12,25%	16,64%
7,0	0,40	28,48%	79,28%	11,50%	16,26%
8,0	0,40	27,36%	78,74%	11,25%	16,26%
9,0	0,40	26,33%	78,20%	11,50%	15,89%
0,5	0,50	18,24%	88,47%	27,00%	77,38%
0,6	0,50	24,69%	89,46%	25,25%	79,63%
0,7	0,50	46,00%	89,37%	29,25%	83,55%
0,8	0,50	60,25%	88,11%	35,50%	85,98%
0,9	0,50	65,57%	86,94%	39,50%	93,83%
1,0	0,50	63,63%	84,95%	74,50%	96,26%
2,0	0,50	32,99%	78,56%	62,75%	90,47%
3,0	0,50	53,48%	81,80%	14,75%	22,43%
4,0	0,50	38,22%	81,53%	13,00%	18,13%
5,0	0,50	30,53%	80,45%	12,50%	16,82%
6,0	0,50	28,69%	79,10%	11,75%	16,45%
7,0	0,50	26,54%	79,01%	11,50%	16,26%
8,0	0,50	25,41%	78,11%	11,75%	16,45%
9,0	0,50	24,28%	77,93%	12,00%	15,89%
0,5	0,60	11,37%	86,31%	24,75%	72,34%
0,6	0,60	16,50%	88,02%	28,25%	75,33%
0,7	0,60	23,77%	88,65%	26,75%	78,88%
0,8	0,60	43,24%	88,11%	34,50%	82,80%

Tabela B.1 – continuação da página anterior

$\alpha$	$q_1$	Botões	Ícones	Áreas de Texto	Caixas de Texto
0,9	0,60	56,76%	86,76%	41,25%	89,53%
1,0	0,60	55,84%	84,59%	74,00%	95,33%
2,0	0,60	53,69%	84,41%	39,50%	56,45%
3,0	0,60	45,49%	81,62%	13,75%	20,19%
4,0	0,60	33,40%	80,99%	13,00%	17,57%
5,0	0,60	29,00%	79,73%	12,25%	17,01%
6,0	0,60	27,05%	79,19%	11,75%	17,38%
7,0	0,60	25,51%	78,47%	11,75%	18,13%
8,0	0,60	24,59%	77,84%	12,00%	17,76%
9,0	0,60	24,28%	77,66%	12,00%	17,38%
0,5	0,70	08,50%	84,68%	31,75%	70,28%
0,6	0,70	12,81%	85,50%	24,75%	72,34%
0,7	0,70	20,39%	86,94%	31,00%	75,51%
0,8	0,70	28,89%	87,21%	34,00%	79,63%
0,9	0,70	41,50%	86,40%	42,00%	79,63%
1,0	0,70	48,46%	84,14%	71,25%	94,21%
2,0	0,70	55,23%	82,97%	24,00%	35,89%
3,0	0,70	41,39%	81,08%	14,00%	20,93%
4,0	0,70	31,45%	79,64%	13,00%	18,88%
5,0	0,70	28,38%	79,46%	12,00%	19,25%
6,0	0,70	26,02%	79,01%	11,75%	18,32%
7,0	0,70	25,51%	78,11%	12,00%	17,94%
8,0	0,70	24,28%	78,02%	12,00%	17,76%
9,0	0,70	24,39%	77,66%	12,00%	18,32%
0,5	0,80	05,84%	78,65%	32,25%	71,96%
0,6	0,80	11,78%	82,43%	35,75%	72,34%
0,7	0,80	19,36%	82,88%	31,00%	74,21%
0,8	0,80	26,13%	84,41%	37,25%	70,47%
0,9	0,80	30,33%	85,32%	39,50%	68,41%
1,0	0,80	38,63%	84,50%	67,00%	89,91%
2,0	0,80	51,64%	82,34%	18,00%	29,72%
3,0	0,80	37,70%	80,54%	13,25%	20,19%
4,0	0,80	30,33%	79,28%	13,25%	19,25%
5,0	0,80	27,36%	78,74%	12,25%	19,44%
6,0	0,80	25,82%	78,56%	12,50%	18,50%
7,0	0,80	25,51%	77,93%	12,25%	18,69%
8,0	0,80	24,28%	78,02%	12,00%	19,25%

Tabela B.1 – continuação da página anterior

$\alpha$	$q_1$	Botões	Ícones	Áreas de Texto	Caixas de Texto
9,0	0,80	24,49%	77,57%	12,00%	20,00%
0,5	0,90	04,10%	73,60%	41,50%	80,93%
0,6	0,90	11,37%	78,74%	46,25%	82,62%
0,7	0,90	20,80%	81,80%	51,50%	76,82%
0,8	0,90	26,33%	82,34%	46,25%	67,66%
0,9	0,90	27,25%	83,06%	46,00%	59,44%
1,0	0,90	28,38%	82,70%	63,75%	77,01%
2,0	0,90	49,49%	81,98%	16,00%	26,73%
3,0	0,90	35,66%	79,73%	11,75%	18,88%
4,0	0,90	29,20%	78,92%	12,75%	19,25%
5,0	0,90	27,05%	78,74%	12,25%	20,19%
6,0	0,90	25,82%	78,20%	12,50%	19,81%
7,0	0,90	25,20%	77,75%	12,25%	22,43%
8,0	0,90	24,28%	78,02%	12,00%	24,49%
9,0	0,90	24,39%	77,57%	12,00%	24,49%
0,5	1,00	17,73%	72,79%	64,00%	78,32%
0,6	1,00	19,26%	77,03%	62,00%	79,25%
0,7	1,00	20,59%	81,17%	62,75%	78,88%
0,8	1,00	20,70%	82,16%	62,50%	75,33%
0,9	1,00	19,16%	80,27%	59,50%	66,92%
1,0	1,00	61,17%	86,40%	55,00%	83,74%
2,0	1,00	39,65%	84,23%	59,00%	76,26%
3,0	1,00	47,44%	84,86%	60,50%	75,70%
4,0	1,00	49,69%	84,86%	60,75%	75,70%
5,0	1,00	50,51%	85,05%	60,75%	75,51%
6,0	1,00	50,82%	85,05%	60,75%	75,14%
7,0	1,00	50,92%	85,05%	60,75%	75,14%
8,0	1,00	51,13%	85,05%	60,75%	75,14%
9,0	1,00	51,13%	85,05%	60,75%	75,14%
0,5	1,10	02,15%	46,49%	51,00%	70,47%
0,6	1,10	06,97%	59,37%	47,50%	76,64%
0,7	1,10	12,19%	68,56%	43,75%	81,50%
0,8	1,10	14,55%	71,80%	38,75%	81,50%
0,9	1,10	11,68%	75,59%	39,50%	79,25%
1,0	1,10	34,63%	86,31%	53,50%	71,59%
2,0	1,10	44,98%	80,81%	14,75%	26,36%
3,0	1,10	33,09%	79,19%	12,25%	20,93%

Tabela B.1 – continuação da página anterior

$\alpha$	$q_1$	Botões	Ícones	Áreas de Texto	Caixas de Texto
4,0	1,10	28,07%	78,65%	13,00%	22,43%
5,0	1,10	26,43%	78,65%	12,75%	25,79%
6,0	1,10	25,61%	78,20%	12,50%	27,29%
7,0	1,10	25,10%	77,66%	12,25%	28,04%
8,0	1,10	25,10%	77,39%	12,25%	26,17%
9,0	1,10	24,90%	77,21%	12,00%	26,17%
0,5	1,20	05,12%	21,44%	54,00%	65,79%
0,6	1,20	10,76%	29,64%	49,00%	69,91%
0,7	1,20	17,93%	44,77%	45,25%	75,89%
0,8	1,20	27,15%	65,23%	41,00%	79,81%
0,9	1,20	33,50%	79,73%	39,75%	79,25%
1,0	1,20	48,36%	86,76%	48,00%	73,46%
2,0	1,20	43,44%	80,81%	14,75%	26,36%
3,0	1,20	32,48%	78,83%	12,25%	22,43%
4,0	1,20	27,97%	78,29%	13,00%	25,42%
5,0	1,20	26,33%	78,38%	12,75%	29,53%
6,0	1,20	26,33%	78,02%	12,50%	29,72%
7,0	1,20	25,72%	77,75%	12,25%	28,79%
8,0	1,20	25,00%	77,30%	12,25%	28,41%
9,0	1,20	24,90%	77,12%	12,25%	26,17%
0,5	1,30	14,24%	23,24%	53,75%	62,62%
0,6	1,30	20,80%	28,29%	51,00%	67,66%
0,7	1,30	28,38%	39,64%	46,75%	74,39%
0,8	1,30	37,70%	63,42%	45,25%	78,69%
0,9	1,30	38,63%	81,71%	43,50%	76,07%
1,0	1,30	49,08%	87,12%	47,25%	70,09%
2,0	1,30	41,70%	80,63%	14,75%	26,54%
3,0	1,30	31,45%	78,38%	12,25%	23,74%
4,0	1,30	27,77%	78,29%	12,75%	30,65%
5,0	1,30	26,95%	78,02%	12,75%	31,03%
6,0	1,30	26,33%	78,02%	12,50%	30,47%
7,0	1,30	25,82%	77,66%	12,25%	29,53%
8,0	1,30	25,00%	77,21%	12,25%	28,41%
9,0	1,30	24,90%	77,03%	12,25%	26,92%
0,5	1,40	22,95%	27,57%	52,00%	62,06%
0,6	1,40	29,41%	31,26%	50,50%	68,22%
0,7	1,40	35,86%	39,55%	47,25%	72,52%

Tabela B.1 – continuação da página anterior

$\alpha$	$q_1$	Botões	Ícones	Áreas de Texto	Caixas de Texto
0,8	1,40	42,11%	58,38%	46,50%	76,26%
0,9	1,40	40,27%	81,17%	44,75%	74,02%
1,0	1,40	47,44%	86,40%	45,00%	67,48%
2,0	1,40	40,27%	80,63%	14,50%	28,41%
3,0	1,40	30,74%	78,20%	12,50%	26,73%
4,0	1,40	27,66%	77,93%	12,75%	32,15%
5,0	1,40	27,05%	78,11%	12,75%	31,59%
6,0	1,40	26,33%	77,93%	12,50%	31,21%
7,0	1,40	25,61%	77,57%	12,25%	29,53%
8,0	1,40	25,00%	77,12%	12,25%	28,41%
9,0	1,40	24,80%	77,03%	12,25%	27,66%
0,5	1,50	27,46%	30,45%	50,75%	60,00%
0,6	1,50	33,91%	34,95%	49,50%	67,29%
0,7	1,50	40,88%	41,98%	49,25%	71,59%
0,8	1,50	41,91%	57,93%	47,75%	73,08%
0,9	1,50	40,47%	79,28%	46,00%	71,03%
1,0	1,50	45,80%	85,50%	43,50%	64,67%
2,0	1,50	39,55%	80,81%	14,25%	28,79%
3,0	1,50	30,64%	78,11%	12,50%	30,47%
4,0	1,50	28,18%	77,84%	12,75%	31,96%
5,0	1,50	26,84%	78,20%	12,75%	32,52%
6,0	1,50	26,43%	77,93%	12,50%	31,96%
7,0	1,50	25,61%	77,48%	12,25%	29,53%
8,0	1,50	24,90%	77,12%	12,25%	28,41%
9,0	1,50	24,80%	76,85%	12,50%	27,66%
0,5	1,60	30,64%	34,77%	49,50%	61,31%
0,6	1,60	37,70%	38,38%	48,50%	66,36%
0,7	1,60	43,44%	43,33%	48,25%	70,65%
0,8	1,60	40,98%	60,27%	46,50%	70,47%
0,9	1,60	40,37%	79,19%	46,00%	68,97%
1,0	1,60	44,57%	84,77%	41,75%	61,87%
2,0	1,60	38,93%	80,45%	14,25%	31,40%
3,0	1,60	30,33%	77,84%	12,75%	30,65%
4,0	1,60	28,38%	77,75%	12,75%	32,34%
5,0	1,60	26,84%	78,20%	12,50%	33,27%
6,0	1,60	26,43%	77,84%	12,50%	31,96%
7,0	1,60	25,31%	77,39%	12,25%	30,28%

Tabela B.1 – continuação da página anterior

$\alpha$	$q_1$	Botões	Ícones	Áreas de Texto	Caixas de Texto
8,0	1,60	24,90%	77,12%	12,25%	28,41%
9,0	1,60	24,80%	76,76%	12,50%	27,66%
0,5	1,70	35,04%	39,10%	48,25%	61,12%
0,6	1,70	39,96%	40,63%	49,25%	65,79%
0,7	1,70	44,06%	45,23%	48,50%	68,04%
0,8	1,70	39,96%	62,61%	46,75%	68,04%
0,9	1,70	40,06%	78,92%	46,00%	66,73%
1,0	1,70	44,26%	84,23%	41,25%	60,19%
2,0	1,70	38,42%	80,00%	14,00%	31,78%
3,0	1,70	30,12%	77,84%	13,00%	31,59%
4,0	1,70	28,38%	77,57%	12,75%	33,08%
5,0	1,70	26,84%	78,11%	12,50%	33,27%
6,0	1,70	26,33%	77,84%	12,50%	32,71%
7,0	1,70	25,31%	77,30%	12,25%	30,28%
8,0	1,70	24,90%	77,12%	12,25%	28,41%
9,0	1,70	24,80%	76,76%	12,50%	27,66%
0,5	1,80	35,96%	41,35%	48,00%	60,37%
0,6	1,80	41,91%	43,24%	50,00%	64,49%
0,7	1,80	44,06%	47,12%	49,50%	66,92%
0,8	1,80	38,83%	64,50%	47,50%	66,73%
0,9	1,80	40,78%	78,20%	44,75%	64,49%
1,0	1,80	43,85%	83,69%	41,00%	58,88%
2,0	1,80	37,50%	79,91%	14,25%	31,21%
3,0	1,80	29,71%	77,93%	13,75%	32,34%
4,0	1,80	28,38%	77,48%	12,75%	33,08%
5,0	1,80	26,84%	78,02%	12,50%	33,08%
6,0	1,80	26,43%	77,84%	12,50%	32,71%
7,0	1,80	25,31%	77,30%	12,25%	30,84%
8,0	1,80	24,90%	77,03%	12,25%	28,41%
9,0	1,80	24,80%	76,76%	12,50%	27,66%
0,5	1,90	37,91%	44,05%	47,00%	60,56%
0,6	1,90	43,14%	45,05%	50,50%	63,74%
0,7	1,90	44,36%	48,92%	49,00%	65,79%
0,8	1,90	38,22%	66,13%	47,50%	65,79%
0,9	1,90	41,19%	78,02%	43,75%	61,50%
1,0	1,90	42,93%	83,06%	40,00%	56,07%
2,0	1,90	36,89%	79,64%	14,25%	32,71%

Tabela B.1 – continuação da página anterior

$\alpha$	$q_1$	Botões	Ícones	Áreas de Texto	Caixas de Texto
3,0	1,90	30,23%	77,75%	13,75%	32,52%
4,0	1,90	28,38%	77,39%	12,75%	33,08%
5,0	1,90	26,84%	77,93%	12,50%	32,52%
6,0	1,90	26,43%	77,84%	12,50%	32,71%
7,0	1,90	25,31%	77,21%	12,25%	30,84%
8,0	1,90	24,80%	77,03%	12,50%	28,41%
9,0	1,90	24,80%	76,76%	12,50%	27,66%
0,5	2,00	34,84%	79,37%	68,75%	89,16%
0,6	2,00	44,98%	45,95%	50,25%	62,99%
0,7	2,00	42,73%	50,36%	47,25%	64,67%
0,8	2,00	37,50%	66,94%	46,75%	64,11%
0,9	2,00	41,19%	77,93%	43,00%	59,81%
1,0	2,00	42,83%	82,52%	38,75%	53,64%
2,0	2,00	36,78%	79,64%	14,25%	33,46%
3,0	2,00	30,23%	77,48%	13,75%	32,52%
4,0	2,00	28,38%	77,57%	12,75%	33,08%
5,0	2,00	26,84%	77,75%	12,50%	32,52%
6,0	2,00	26,43%	77,84%	12,50%	32,71%
7,0	2,00	25,31%	77,21%	12,25%	30,84%
8,0	2,00	24,80%	77,03%	12,50%	28,41%
9,0	2,00	24,80%	76,76%	12,50%	27,66%

## Apêndice C

# PARÂMETROS UTILIZADOS

Abaixo segue uma lista com valores e constantes utilizados para os parâmetros das fórmulas apresentadas neste trabalho.

Parâmetro	Símbolos e Valores
Constante de Boltzmann	$k_B = 1,38062 \cdot 10^{23}$ joules/kelvin
Constante entrópica	$q$ varia entre 0,01 e 2,00 para $q = 1$ , tem-se o valor da entropia de Shannon
Relação entre $q_1$ e $q_2$	$\alpha = \frac{q_1}{q_2}$ varia entre 0,5 e 9,0. Para $\alpha = 1$ , $q_1 = q_2$ , tornando a Equação da entropia não-extensiva estendida igual à forma de Tsallis
Ponto de separação do histograma	$t$ , quando definido manualmente, geralmente é utilizado o valor 127. No algoritmo limiar adaptativo, o valor de $t$ varia ao longo da imagem. A segmentação por meio de entropia não-extensiva calcula o valor de $q$ com base na quantidade de informação das duas distribuições que são formadas após a separação do histograma

# Livros Grátis

( <http://www.livrosgratis.com.br> )

Milhares de Livros para Download:

[Baixar livros de Administração](#)

[Baixar livros de Agronomia](#)

[Baixar livros de Arquitetura](#)

[Baixar livros de Artes](#)

[Baixar livros de Astronomia](#)

[Baixar livros de Biologia Geral](#)

[Baixar livros de Ciência da Computação](#)

[Baixar livros de Ciência da Informação](#)

[Baixar livros de Ciência Política](#)

[Baixar livros de Ciências da Saúde](#)

[Baixar livros de Comunicação](#)

[Baixar livros do Conselho Nacional de Educação - CNE](#)

[Baixar livros de Defesa civil](#)

[Baixar livros de Direito](#)

[Baixar livros de Direitos humanos](#)

[Baixar livros de Economia](#)

[Baixar livros de Economia Doméstica](#)

[Baixar livros de Educação](#)

[Baixar livros de Educação - Trânsito](#)

[Baixar livros de Educação Física](#)

[Baixar livros de Engenharia Aeroespacial](#)

[Baixar livros de Farmácia](#)

[Baixar livros de Filosofia](#)

[Baixar livros de Física](#)

[Baixar livros de Geociências](#)

[Baixar livros de Geografia](#)

[Baixar livros de História](#)

[Baixar livros de Línguas](#)

[Baixar livros de Literatura](#)  
[Baixar livros de Literatura de Cordel](#)  
[Baixar livros de Literatura Infantil](#)  
[Baixar livros de Matemática](#)  
[Baixar livros de Medicina](#)  
[Baixar livros de Medicina Veterinária](#)  
[Baixar livros de Meio Ambiente](#)  
[Baixar livros de Meteorologia](#)  
[Baixar Monografias e TCC](#)  
[Baixar livros Multidisciplinar](#)  
[Baixar livros de Música](#)  
[Baixar livros de Psicologia](#)  
[Baixar livros de Química](#)  
[Baixar livros de Saúde Coletiva](#)  
[Baixar livros de Serviço Social](#)  
[Baixar livros de Sociologia](#)  
[Baixar livros de Teologia](#)  
[Baixar livros de Trabalho](#)  
[Baixar livros de Turismo](#)